UiO : University of Oslo

Vinit Ravishankar

Understanding Multilingual Language Models

Training, Representation and Architecture

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Sciences

2023

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo

ISSN

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented here was conducted at the University of Oslo, under the supervision of professor Lilja Øvrelid and professor Erik Velldal. A part of this research was also conducted on a research visit to the University of Copenhagen, under the supervision of professor Anders Søgaard.

The thesis is a collection of nine papers, presented non-chronologically. In addition, the thesis consists of an introductory section, followed by three chapters providing the technical background necessary to contextualise these papers, followed by a conclusion.

Acknowledgements

It is impossible for me to properly thank and acknowledge everyone I could rely on in just a few short paragraphs. In the end, it is hard for me to think of this thesis as a solely individual accomplishment; it would never have materialised without the encouragement and support of so many people, near and far, over the course of the past four years.

I am grateful to my supervisors, Lilja and Erik, who were immensely supportive when progress seemed slow, and always encouraged me to pursue my research interests. I am also thankful to Anders for hosting me for more than a year, giving me the opportunity to broaden my horizons. To Joakim, who also provided me with solid mentorship, I am grateful. The LTG and CoAStAL have both been immensely intellectually and socially stimulating environments, and I am grateful to all members of both, past and present.

On a personal note, this thesis owes a lot to Artur and Mostafa, for our almost daily conversations that have considerably shaped a lot of the decisions – both good and terrible – that I have made over the past four years. This thesis would also likely have been a lot more unhinged without both of them, as well as Sachit, Nick and Yamini, who read and commented on my initial drafts, resulting in an altogether more measured work. I am also grateful to Nora, whose support I could always rely on, especially during the darkest winter of the pandemic. Finally, I am grateful to my family, and to all my friends, old and new, scattered all over the world – for keeping me sane, despite my proximity to the Arctic Circle.

Vinit Ravishankar
Oslo, April 2023

Summary

The field of natural language processing, or NLP, has seen numerous paradigm shifts since its inception, as machine translation, during the Cold War. Each such shift – from rule-based methods to statistical ones; from statistical models built on sparse, selected features to those built on dense, learnt features; from simpler neural networks to highly parameterised, deep, pretrained models – has been characterised by spurts of increased research productivity. The current dominant paradigm, that of deep pretrained language models, has been accompanied by precisely such an increase in research productivity, divided over a number of subfields of research.

We position this work in the intersection of two such subfields. The first of these pertains to the interpretability of language models, or on methods attempting either to describe model behaviour, or to find explanations for model behaviour; the field includes methods that, for instance, attempt to characterise and describe the information a model has acquired, through being trained on large corpora. The other subfield is that of multilinguality, or that of a model's capacity to learn to model and process textual language in multiple languages. Often, multilinguality is enabled simply by training models on large corpora consisting of multilingual text. This thesis is composed of nine papers that attempt to both raise and answer questions situated in this intersection. In order to do so, we adopt a series of lenses, or analytical frameworks, through which we analyse neural models of language.

The first of these lenses is that of examining the quality of the multilingual spaces that emerge during training. First, we analyse the effect that small amounts of language-specific fine-tuning data has on these language models; next, we analyse the role that the ratio of language data in different languages plays on model performance; finally, we analyse the effect of typological features in our corpora, upon multilingual model performance.

Next, we analyse our models through a family of analytical models called probes, which we use in an effort to describe what kind of grammatical information can be extracted from our models. First, we analyse sentence encoders transferred from English to other languages via transfer learning; next, we apply the same analysis to deep multilingual language models; finally, we attempt to extract syntactic trees from language models, quantifying how differences in syntactic formalism affect this extractability.

Our last lens involves the examination of a model's internals; we specifically focus on the highly popular transformer architecture, and attempt to quantify the effect different components have on language learning. First, we analyse the extent to which transformer attention weights can store syntactic information across languages; next, we analyse the effect that the choice of position embedding method has on multilingual space quality; finally, we analyse the capacity of language models to learn from scrambled text, and the role that position embeddings play in imparting order to language models.

Thus, over the course of this thesis, we first provide the reader with a summary of the state of the research relevant to our work, from multilinguality

in language models, to the principles behind probing, to descriptions of the transformer mechanism and the debate surrounding the interpretability of attention; in each such chapter, we attempt to contextualise our own work. We proceed to conclude with an examination of the state of NLP, particularly relevant to our contribution, both along technical and societal lines. Thus, we describe potential future research avenues that could emerge from our work, as well as the implications of research into multilinguality and interpretability on society.

Sammendrag

Fagfeltet språkteknologi, "natural language processing", eller NLP, har gjennomgått en rekke paradigmeskifter siden det oppsto under den kalde krigen med målet om å utvikle maskinoversettelse. Hvert slikt skifte – fra regelbaserte metoder til statistiske metoder, fra statistiske modeller basert på manuelt utvalgte trekk, til de som er bygget på lærte trekk, fra enklere nevrale nettverk til svært parametriserte, dype pre-trente ("pretrained") modeller – har vært karaktisert av en en bølge med økt forskningsproduktivitet. Det nåværende dominerende paradigmet, som kjennetegnes av dype språkmodeller, har blitt ledsaget av nettopp en slik økning i forskningsproduktivitet, fordelt på en rekke underfelter innenfor forskningen. Vi plasserer dette arbeidet i skjæringspunktet mellom to slike underfelt. Den første av disse gjelder tolkbarhet ("intepretability") av språkmodeller, eller metoder som forsøker å enten beskrive modellatferd, eller å finne forklaringer på modellatferd. Feltet omfatter metoder som for eksempel forsøker å karakterisere og beskrive informasjonen en modell har tilegnet seg, gjennom å bli trent på store tekstmengder. Det andre underfeltet er flerspråklighet, eller modellens evne til å lære å modellere og behandle tekst på flere språk. Ofte aktiveres flerspråklighet ganske enkelt ved å trene modeller på store tekstmengder bestående av flerspråklig tekst. Denne avhandlingen er sammensatt av ni artikler som forsøker å både stille og besvare forskningsspørsmål plassert i dette skjæringspunktet. For å gjøre det tar vi i bruk flere linser, eller analytiske rammer, som vi analyserer disse nevrale språkmodellene gjennom.

Den første av disse linsene består i å undersøke kvaliteten på de flerspråklige egenskapene som dukker opp under trening. Først analyserer vi effekten som små mengder språkspesifikke data bruke til såkal "fine-tuning" av modellene har på språkmodellene. Deretter analyserer vi hvilken rolle forholdet mellom data på ulike språk og modellens ytelse. Til slutt analyserer vi effekten av typologiske trekk i våre datasett på flerspråklig modellytelse.

Deretter analyserer vi modellene våre gjennom en familie av analytiske modeller ("probes"), som vi bruker i et forsøk på å beskrive hva slags grammatisk informasjon kan trekkes ut fra modellene våre. Først analyserer vi setningsrepresentasjoner fra såkalte "sentence encoders" overført fra engelsk til andre språk via transfer-læring. Deretter appliserer vi den samme analysen på dype flerspråklige språkmodeller; til slutt undersøker vi muligheten for å trekket ut syntaktiske trær fra slike språkmodeller, og kvantifiserer hvordan forskjeller i syntaktisk formalisme påvirker resultatene.

Vår siste linse innebærer undersøkelse av disse modellenes indre. Vi fokuserer spesielt på den svært populære transformer-arkitekturen, og forsøker å kvantifisere effekten av ulike komponenter på språklæring. Først undersøker vi i hvilken grad transformerens oppmerksomhet ("attention") kan lagre syntaktisk informasjon på tvers av språk. Deretter analyserer vi effekten som valget av posisjonsembedding har på kvaliteten av flerspråklige egenskaper; til slutt analyserer vi disse språkmodellenes kapasitet til å lære av tekst som ikke følger språkets leddstilling, og rollen som posisjonsembeddings spiller for å legge til rette for gode språkmodeller.

I introduksjonen til denne oppgaven gir vi derfor først leseren en oppsummering av forskningen som er relevant for vårt arbeid, fra flerspråklighet i språkmodeller, til prinsippene bak probing, til beskrivelser av transformerarkitekturen og debatten rundt tolkbarhet av oppmerksomhet. I hvert introduksjonskapittel prøver vi derfor å kontekstualisere vårt eget arbeid. Vi avslutter med en undersøkelse av den nåværede tilstanden til forskningsfeltet NLP, som er særlig relevant for oss, både langs tekniske og samfunnsmessige linjer. Dermed beskriver vi potensielle fremtidige retninger som kan springe ut fra denne forskningen, samt implikasjonene av forskning på flerspråklighet og tolkbarhet av nevrale modeller på samfunnet som helhet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language processing (NLP), or the computational processing of human languages, traces back to an era not long after the invention of computers; the field could be said to have its origins sometime around the Cold War, which led to a sharp interest in machine translation systems. These systems were initially based on labouriously devised rules; eventually, statistically-driven methods began to dominate. The 2010s saw the emergence of deep learning as a paradigm, replacing the earlier manual feature-driven statistical models: it is in the midst of precisely this paradigm that we find ourselves situated. One might also say that the history of NLP has been full of boom and bust cycles; today, in early 2023, we find ourselves in the midst of a massive boom cycle.

The rise of modern, deep-learning based methods for NLP has led to the emergence of the two broad threads of research that we attempt to unify in this thesis. The first of these is the *multilinguality* of NLP systems, or their capacity to learn to process multiple languages, simultaneously or otherwise. Given the dominance of English as the world's *lingua franca*, and given that that a substantial chunk of data available on the internet is in English, English-language NLP tends to dominate both the research and commercial landscape. Numerous approaches to creating and enabling more multilingual language models have been proposed: these include, for instance, methods that take advantage of the amount of English data available to learn some notion of *language* (in the general sense), and then use more limited data to learn to process *a* language (in the specific sense). This multilinguality is often discussed in the context of *language models* (LMs), which are large machine learning models trained to predict words, given their context. Language models have demonstrated substantial utility in their capacity for adaptation to other tasks, once trained in such a fashion. There is much we do not fully understand about the nature of multilingual models: while training scenarios are often similar to the monolingual context, and involve simply training large models on multiple languages, the models often display unexpected characteristics, such as some ability to translate words from one language to another, despite never explicitly being trained to do so.

This brings us to our second observation – we often do not fully understand what our models are doing when they arrive at a particular decision (whether in a monolingual context or multilingual), or why their parameters look they way they do. Conventional feature-based methods could, for instance, give us direct feature weights, allowing humans to trivially diagnose the importance of different factors in model behaviour[1] down to model behaviour on a single

[1] Note that while we use the term *behaviour* throughout this thesis, we do so with no intent to anthropomorphise language models.

instance; with deep neural methods, however, it is not entirely straightforward what these 'features' even are, as they are learnt rather than selected, and dense rather than sparse. Numerous explainability methods have been proposed, each with their own set of flaws, or limitations in terms of use cases. Further, while a considerable amount of research focus has been dedicated to analysing and interpreting these models, only a fraction of this focus has been diverted to the multilingual context.

The focus of this thesis is therefore to contribute to the literature on the intersection of these research domains. Narrowing down somewhat, the core aim of this thesis is to contribute to the analysis of multilinguality in language models; in order to do so, we adopt a variety of "lenses" (§1.1), i.e. over the course of this work, divided into nine research papers, we tackle this broad domain from different angles. Thus, we experiment with alterations in training data to see how it affects the degree and nature of multilinguality our models are capable of; we probe representations generated by multilingual language models to see what forms of linguistic information they can encapsulate; finally, we experiment with specific components within transformer-based language models, to narrow down the role these components play in the context of multilinguality.

All in all, the goal of this thesis is not to provide an overarching answer to how multilingual language models function: indeed, it is unclear how meaningful such a grand "theory of everything" could even be for models with billions of parameters, when an overarching *question* is difficult to formulate in and of itself. Instead, over the past four years, we chose to focus on asking very specific questions (§1.2), with the goal of contributing to the understanding of multilinguality in LMs, and on devising experiments to attempt to answer these questions.

1.1 Analytical lenses

Having established the broad domain of this thesis – interpreting the behaviour of language models in multilingual contexts – we proceed to define three such interpretative lenses; we structure the narrative of this thesis along the lines of these lenses.

Multilingual spaces This strand of research involved examining how specific corpus characteristics could affect the creation or modification of a model's multilingual space; that is, how the nature of the multilingual corpus would reflect in the behaviour of the multilingual model itself. Thus, in Chapter 2, we first present a brief history of deep language models; we then shift focus to the emergence of multilinguality in NLP discourse, the training of multilingual models, and the analysis of their multilingual spaces, fitting three of our papers into this context.

Probing Next, we analyse language models by *probing* them, 'probing' referring to the application of analytical methods that attempt to extract information from language model representations of words or sentences. Probing methods often use some sort of system – such as classifiers – built on top of existing language models, and trained on datasets consisting of some sort of informative linguistic phenomenon. Downstream performance on these datasets is meant to encapsulate the extent to which language models have learnt the relevant phenomenon. In Chapter 3, we discuss the insights and issues surrounding probing, as well as recent innovation in the field; we contextualise our work in light of this background.

Architectural analyses Finally, we zoom into our language models and start to analyse their internals: thus, in Chapter 4, we begin with a detailed descriptions of the transformer, perhaps the class of language model most relevant to NLP at the time of writing this thesis. We focus on a select subset of transformer components, and describe their contribution to modelling (pseudo)-multilingual language. In this light, we further address the role and importance of word order in language model pretraining.

1.2 Research questions and contributions

Given these three lenses, we now outline a set of research questions, each corresponding to a lens.

1. What can we say about the influences that differences in training or fine-tuning corpora have on the quality of multilingual language model spaces?
   - What effect does fine-tuning on a few annotated instances in a target language have upon zero-shot transfer results?
   - What role do differences in per-language corpus size play on multilingual performance?
   - Are there cross-linguistic patterns determining how easy bootstrapping multilingual spaces can be?
2. What linguistic properties can we extract from multilingual language models?
   - How do previously established linguistic probing tasks hold up in multilingual settings, given multilingual sentence encoders?
   - How do these same multilingual results look when evaluated on deep multilingual language models?
   - How do differences in syntax formalisms affect how easy they are to extract from a language model, across languages?

3. How do specific components within transformer-based language models act in multilingual contexts?
    - What effect does fine-tuning on annotated syntactic corpora have upon attention weight distributions?
    - What role does the choice of token position representation method play on multilingual performance?
    - How is the performance of models trained on scrambled corpora different to that of ordinary language models, and why?

1.3 Summary of papers

Finally, we flesh out our actual contribution along these lenses and research questions – that is, the papers that constitute this thesis. Each paper attempts to answer a research question, and each group of papers corresponds to a lens. Note that the order of presentation of these papers is *not* chronological.

I In Lauscher et al. (2020), we describe the performance of language models in the context of *few-shot* fine-tuning. Precisely speaking, in this work, we quantify the gap between language model performance on a variety of tasks in the context of zero-shot transfer, and few-shot transfer – i.e., transfer learning given fine-tuning corpora with very few annotated instances.

I was the shared first author on this paper; all authors contributed equally to the idea behind the paper, and my co-first author and I split the experimental setup; I worked on the lower level tasks. I also created the plots and figures used in this paper.

II In Ravishankar et al. (2021b), we train multiple autoregressive language models on multilingual training corpora, and evaluate the difference in LM performance given varying distributions of the languages that constitute the training corpus.

I was the principal author on this paper.

III Finally, we conclude this section with a study of the inductive biases present in multilingual space construction (Ravishankar and Nivre, 2022), and discuss how language features – both broader typological features, but also corpus-level statistics – affect the quality of multilingual spaces built during pre-training.

I was the principal author on this paper.

IV We begin our series of probing-related experiments with Ravishankar et al. (2019c), where we describe an approach to probing language models trained in the context of *cross-linguistic transfer learning*; in doing so, we attempt to isolate the linguistic phenomena that language models may retain when their representations are transferred to a non-English language.

I was the principal author on this paper.

V In Ravishankar et al. (2019b), we extend our experiments from our previous paper; we take into account the rapidly changing landscape of NLP, and evaluate deep multilingual pretrained language models to assess what specific linguistic phenomena they may retain.

I was the principal author on this paper.

VI Concluding our series of probing-related experiments, in Kulmizev et al. (2020a), we attempt to extract syntactic structure from language models, by means of principled probes. We contrast two different formalisms used to denote dependency syntax, in an attempt to determine whether or not language models could have formalism 'preferences'.

I was the second author on this paper; my contribution was discussing the design of the experimental setup, and generating data for analysis.

VII Bridging the gap between the corpus-related experiments and architectural analyses, in Ravishankar et al. (2021a), we use an established method to extract syntactic structure from the attention mechanism weights of a pretrained language model. We contrast the quality of this extracted structure across languages, and show that the quality of extracted trees sharply improves when language models are fine-tuned on small dependency-annotated corpora. We analyse the relevance of different language model components to this improvement in tree extraction by freezing components prior to fine-tuning.

I was the shared first author on this paper. I was responsible for the entire second half of this paper, i.e., for the sections on analysing the effects of fine-tuning.

VIII Given the clear dominance of transformer-based language model architectures in the literature, we attempt to isolate the influence of specific transformer architecture components on multilingual training. We focus, in Ravishankar and Søgaard (2021), on positional representations, and show that more complex recent innovations in positional representation may underperform in the multilingual context.

I was the principal author on this paper.

IX Concluding our architectural analyses, in Abdou et al. (2022), we address a thread in the literature discussing language model performance when pretrained on scrambled text – which allegedly tends to be surprisingly good. We quantify the extent to which masked language models learn word position via position embeddings, and isolate several word order 'clues' present in scrambled corpora; finally, we demonstrate that the more complex the downstream task, the more critical word order signals become.

I was the shared first author on this paper. My contributions involved determining hidden word-order signals in language models, and the analyses of the model's attention. I also designed all figures in the paper.

The next three chapters involve a close look at the three research threads we outlined earlier: multilingual spaces in Chapter 2, probing in Chapter 3, and the internal behaviour of transformers in Chapter 4. We conclude this thesis with an examination of the scientific and social implications relevant to this work in Chapter 5. The nine papers that we include as part of this thesis are attached following our conclusion; shared authorship is indicated with an asterisk.

Chapter 2

Language models and multilinguality

We begin this review by briefly summarising the paradigm shifts in NLP research that have taken place over the past decade or so. This section is meant to be architecture-agnostic: we describe, first, the emergence of deep learning for NLP, and then branch into describing the dominance of deep pretrained language models, as well as perspectives on enabling multilingualism in NLP. We defer concrete discussions on the *architecture* of specific language models to Chapter 4. As such, this section involves the background relevant to our work on analysing (multilingual) language model behaviour over the training or fine-tuning *process* – two terms that we elaborate upon later in this section.

2.1 A brief history

Traditionally, the statistical models that replaced rule-based NLP systems would often operate on word *features*, represented by sparse vectors indicating the value of a particular feature (eg. a part-of-speech tag). These statistical models could be stacked, or pipelined: features for "higher-level" semantic tasks could be derived from the outputs of "lower-level" models upstream. The spread of deep-learning based methods led to the gradual phasing out of these sparse feature vectors, and their replacement with automatically learnt *dense* features.

2.1.1 From sparse to dense features

A significant innovation in modern (deep) NLP came from the use of dense feature vectors. The use of distributional features in NLP broadly stems back to Firth's oft-repeated claim (1957): *you will know a word by the company it keeps*. Distributional methods in NLP involved learning and representing token features as vectors in a continuous feature space (Bengio et al., 2000), based on the contexts in which a particular token was likely to appear. Thus, tokens like *apple* and *orange*, which are likely to occur in similar contexts, would end up with dense representations that were closer in vector space than *apple* and *aardvark*.

Many early methods for learning these representations were computationally expensive, and the availability of computational power remained a bottleneck for further adoption. Mikolov et al. (2013a) made significant contributions to efficiency by describing the continuous *bag-of-words* (CBOW) and *skipgram* models: two word embedding models with log-linear complexity, dubbed word2vec. Briefly, these methods learn word embeddings by attempting to either predict a word given its context, or to predict a word's context, given the word itself. The ability to rapidly bootstrap word embedding models from unlabelled text corpora (and to load and save these models) led to the widespread adoption of distributional word representations in NLP. Thus, systems such as classifiers would be built on top of these embedding models, and word representations would replace the previously popular sparse feature vectors. Other methods for generating word embeddings contemporaneous with word2vec include GloVE (Pennington et al., 2014), where the authors use co-occurrence frequency information, and fastText (Bojanowski et al., 2017), where the authors construct representations from character n-grams, enabling generalisation to unseen tokens.

The success that followed from representing words as dense vectors led to research into the logical next step – that of representing *sentences* as dense vectors. An advantage of processing entire sentences is also that this allows for taking word context into account. The 'static' nature of word embeddings has several disadvantages: the same token ought to have different representations in different contexts, something static embeddings cannot easily capture. Further, word senses present another layer of confusion: the words *bank* (as in a financial institution) and *bank* (as in a river bank) ought to have very different representations; however, given that static word representations are essentially lookup tables matching strings to vectors, they are incapable of distinguishing between these senses. Several attempts to address these issues emerged; Kiros et al. (2015) describe *skip-thought vectors*, where the authors use recurrent neural networks (RNNs) to encode a sentence; they use the last state of these RNNs to decode the next and previous sentences. Conneau et al. (2017) experiment with training sentence representations on natural language inference (NLI), by virtue of NLI being seen (at the time) as a fairly complex task, requiring syntactic and semantic knowledge[1].

We refer the reader to a blog post[2] we had co-authored in the first year of this PhD for a more detailed summary of sentence representation methods.

2.1.2 Towards contextualisation, and deep language models

With the release of the ELMo model (Peters et al., 2018b), discourse around sentence representations shifted towards discussing deep, contextualised embedding models. ELMo, specifically, was a bidirectional recurrent neural network (biRNN) trained on a next-token prediction task. The authors use the weights of the model itself as a 'feature generator' of sorts downstream: the two-layer RNN thus outputs word embeddings that take into account the context of a word, allowing these representations to be used by higher layers of the network. They show that adding ELMo as a feature generator enabled them to reach state-of-the-art performance on a broad range of tasks. A critical contribution of this paper was also the fact that language modelling, or predicting the next word in a corpus, was a *semi-supervised task*; such methods,

[1] We now know this to no longer necessarily be the case: see Poliak et al., 2018.

[2] https://supernlp.github.io/2018/11/26/sentreps/

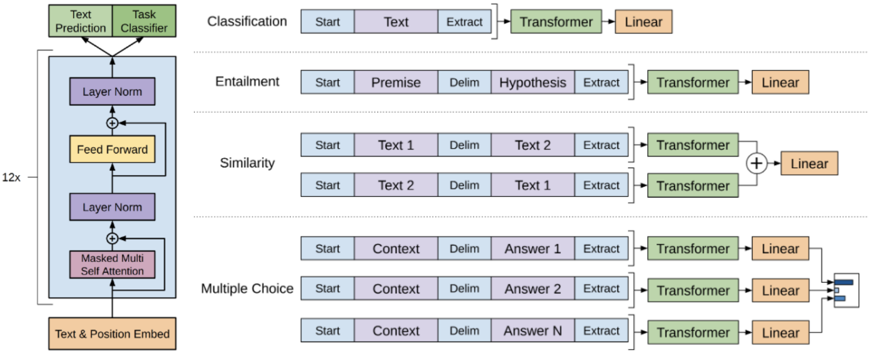


Figure 2.1: GPT's training and evaluation setup on a range of tasks. Image taken from Radford et al. (2018).

based on training on unlabelled data, are inherently more scalable, since no manual human annotation is necessary.

The success of ELMo led to substantial research into the use of pre-trained language models on downstream tasks. Radford et al. (2018) claimed that having to train entire model architectures on top of ELMo representations was too task-specific; in their paper, they describe their training of a large language model, called the Generative Pretrained Transformer (GPT), based on a transformer decoder[3], and demonstrate the utility of this system, given structured input converted into an ordered sequence (see Figure 2.1 for illustrations of how different tasks can be converted to the appropriate model input.). This approach represented a further paradigm shift away from using deep pretrained models as *feature generators*, to using them as systems in and of themselves, with relatively small and uncomplicated task-specific layers being fit to the models. The success of the GPT model rapidly led to the release of two follow-up models, GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) (and eventually the very recent ChatGPT, that we discuss in the conclusion to this work).

Relevant to much of our work is BERT (Devlin et al., 2019), a transformer *encoder*-based language model, wherein several innovations to the task of language modelling were introduced. A key such innovation – one of the cornerstones of architectures based on BERT – was the introduction of *masked language modelling* as a task, contrasted with the autoregressive, sequential, next-token-prediction task from ELMo/GPT. Masked language modelling involved replacing the task of predicting the next token in a given context, with predicting an arbitrarily selected masked token, given every other token in a sequence window. Figure 2.2 is a visualisation of some such word contexts for various approaches to language modelling. Devlin et al. introduce several other innovations, such as *WordPiece tokenisation* (Wu et al., 2016), where a

[3] Note that the specifics of the transformer architecture are not strictly relevant to this section; we discuss transformers in greater detail in Chapter 4

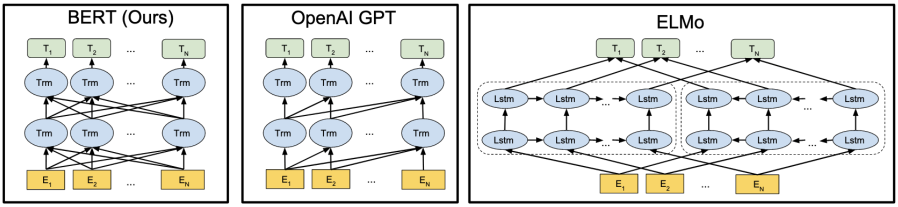


Figure 2.2: Relevant contexts for language modelling with different language models. Image taken from Devlin et al. (2019)

tokenisation schema can be learnt over a large corpus, given a fixed vocab size: the algorithm starts with single characters, and proceeds to merge them in a fashion that maximises the likelihood of the training data[4]. Further, the authors propose using a *next-sentence prediction* task: in addition to predicting masked tokens, the model would learn to predict, given a pair of sentences, whether the second sentence logically followed the first in their source corpus or not.

For a large part of our PhD, BERT was the primary language model that underwent technical analysis, as part of a subfield that came to be known as *BERTology* (Rogers et al., 2020). This is not necessarily still the case, though many popular language models today still happen to be based on innovations applied to BERT's architecture.

2.2 Building multilingual models

Early NLP had mostly been focused on widely-spoken, well-resourced and politically hegemonical languages. In recent years, particularly given the increasing importance of the amount of available training data, the question of how training data for one language can best be utilised to augment training in another more poorly resourced language, has become a dominant strand of research. Somewhat cynically, this outreach has some of its roots in the market utility of targeted advertising in markets that are (as yet) relatively unintegrated into the global economy, such as large parts of Africa, India, and south-east Asia: we discuss this in greater detail in our conclusion in Chapter 5. In the following section, we briefly summarise the approaches to enabling multilingualism in language models.

2.2.1 In the shallow era

Early works focused on multilingualism in word embedding spaces; some attempted to induce multilingualism in a supervised fashion, while others attempted to align embedding spaces in different languages without supervision.

[4] This, in principle, is very similar to the older *byte-pair encoding* (BPE) (Sennrich et al., 2016), the difference being that BPE merges are based on the frequency of potential symbol pairs.

An early such work was Mikolov et al. (2013b), where the authors train embedding models in two languages and learn linear mappings between them to translate at a word level; we refer the reader to Artetxe et al. (2018b) for a summary of supervised word embedding transfer methods.

Moving on to unsupervised approaches, Zhang et al. (2017) describe an adversarial approach to embedding alignment, where they train a generator to transform source language word embeddings to match target language spaces, thereby attempting to fool a discriminator trained to guess what space the embeddings were drawn from. Lample et al. (2018b) describe the contemporaneous system MUSE; where they add an iterative training approach to the discriminator. Artetxe et al. (2018a) describe vecmap, where they use a fully unsupervised weak initialisation, taking advantage of the (idealised) isometry of two embedding spaces, followed by self-supervised optimisation.

There are some limitations to the unsupervised approach: for one, the assumption that embedding spaces in different languages are isometric does not necessarily hold true. Vulić et al. (2019) discuss how, given 15 typologically diverse languages, for 87 out of the 210 possible language pairs, bilingual dictionary induction fails completely when mapped without supervision. In a follow-up work (Vulić et al., 2020a), they show that the non-isometricity of word embedding spaces can also be due to insufficient training data or poor training regimes for monolingual embeddings. Søgaard et al. (2018) analyse the impact of various factors, such as typological or domain differences, on unsupervised cross-linguistic word embeddings; they propose using identically spelled words in both corpora as supervised 'seeds' instead of fully unsupervised methods.

Beyond the word level, Conneau et al. (2018b) describe a transfer approach, wherein they pretrain an English sentence encoder and a classifier on an NLI dataset; they then use a separate parallel corpus to minimise the distance between their English encoder and an encoder for some other language; they then combine the non-English encoder with the English classification layer. Artetxe and Schwenk (2019) describe LASER, where they followed a neural machine translation-style setup, and train bidirectional LSTM models, with shared BPE vocabularies, on parallel corpora for 93 languages; they use English and Spanish as their target languages, and retain only the sentence encoder downstream. Artetxe et al. (2018c) and Lample et al. (2018a) describe approaches to building machine translation systems starting from word-level parallel embeddings.

We refer the reader to Ruder et al. (2019) for a more detailed survey of cross-lingual embedding models.

2.2.2 Multilingualism without transfer

The approaches that we have discussed in the previous section were very quickly superseded by the explicit training of deep multilingual models – almost contemporaneous with the training of deep language models itself. This approach was, in principle, simple to grasp: deep multilingual language models were simply trained on corpora consisting of data in multiple languages.

One of the first such models was a multilingual BERT variant (mBERT) (Devlin et al., 2019), where the authors train a BERT model on the 100 languages with the largest Wikipedias, using Wiki dumps as training data, and a shared WordPiece vocabulary of size 110k. They use a form of exponentially smoothed weighting of corpora to ensure that overrepresented languages were undersampled; i.e. if the probability of a token being sampled from language i were p_i, they adjust probabilities such that $q_i = {}^{p_i^\alpha}/\sum_{j=1}^{N} p_j^\alpha$. We experiment with the influence this weighting α has on multilingualism in Paper II.

mBERT proved to be extremely effective on downstream multilingual tasks; this effectiveness led to considerable innovation in training routines, or in model architecture. The XLM model (Lample and Conneau, 2019) presented some innovations over the vanilla mBERT approach; the authors add *translation language modelling* as an additional task, wherein a sentence pair can be sampled from a sentence and its translation in another language, allowing the MLM step to use both the source and the translation, thereby pushing representations across languages closer together. XLM-RoBERTa (Conneau et al., 2020), unrelated conceptually to XLM, was trained on Common Crawl dumps of very large corpora, albeit only on masked language modelling. The authors discuss the *curse of multilinguality*: for a certain model capacity, while cross-lingual performance tends to improve with the addition of more languages, this is only up to a certain point, past which both monolingual and cross-lingual performance begin to degrade. The authors show that a simple increase in model capacity – i.e., in the number of model parameters – helps alleviate this; hence their emphasis on 'learning at scale'.

2.2.3 Zero-shot learning

A large part of the success of multilingual language models is due to their ability to leverage *transfer learning* for downstream tasks, an approach often dubbed *zero-shot learning*, in its most resource-sparse scenario. Zero-shot learning involves taking pre-trained language models and training additional layers – for instance, classification heads – on top of them, for a particular task in a particular language (typically English). This may or may not imply the absence of fine-tuning; i.e. the parameters of the model itself may or may not be frozen. Downstream, the multilingual space within the model ensures that representations for different languages are still somewhat similar; the result is that when evaluated on an entirely different language, these systems still show competitive performance. This is exceedingly useful for languages where large, annotated, task-specific training corpora do not exist, and state-of-the-art approaches to a variety of multilingual tasks continue to use zero-shot learning.

An advantage of the zero-shot learning scenario is also that it is considerably less expensive to bootstrap evaluation datasets for underresourced langauges, than it is full training datasets. Typically, a model's capacity to generalise cross-linguistically is seen using its zero-shot performance on these evaluation datasets, using specific downstream tasks as a lens; this, of course, has its own set of issues, something we address more generally in §4.4.2. Relevant datasets include the Universal Dependencies project (Nivre et al., 2017), a multilingual dataset of treebanks with part-of-speech (POS) and dependency syntax annotations in over 100 languages; WikiANN (Rahimi et al., 2019), a multilingual named entity recognition (NER) corpus; PAWS-X (Yang et al., 2019), a dataset for paraphrase identification in 6 languages; XNLI (Conneau et al., 2018b), a natural language inference corpus, with dev and test splits in 14 languages; XQuAD (Artetxe et al., 2020), a question-answering dataset, with dev partitions in 11 languages. XTREME (Hu et al., 2020) is a large-scale dataset of 9 syntactic and semantic tasks, in (a total of) 40 languages.

In Paper I, we experiment with *few-shot* learning scenarios, wherein a small number of annotated instances are provided to the model to learn from; we demonstrate and quantify its utility over zero-shot learning in the context of a variety of NLP tasks.

2.3 Multilingual spaces

An important lens for the analysis of embeddings, both mono- and multilingual, is to analyse how these embedding spaces actually look – i.e. how embedding vectors are distributed within continuous vector space. As such, a strand of research that emerged in light of the dominance of massive multilingual models was that of describing the nature of this multilingualism: this involved attempting to quantify the extent to which these models are multilingual at all, and attempting to determine precisely how they *become* multilingual.

2.3.1 How multilingual are these spaces?

An early work in this direction was Pires et al. (2019); the authors analyse mBERT's ability to generalise cross-linguistically for NER and POS tagging, and show that transfer is best when the languages are typologically similar, and that mBERT is still capable of generalisation even for languages written in different scripts (implying zero lexical overlap). They hypothesise that this could be due to shared subword ‘anchors’: URLs, numbers etc. A contemporaneous work was Wu and Dredze (2019); the authors experiment with a wider array of tasks, and show that mBERT often outperforms state-of-the-art models pretrained with cross-lingual signals. They run a set of further experiments and show that despite impressive cross-lingual performance, all of mBERTs layers retain language-specific information, and that there exists a strong correlation between the frequency of shared anchors and transfer performance. In a follow-up work (Wu et al., 2020a), however, they show that the effect of these anchors was blown out of proportion, and that *parameter sharing* across languages enables effective transfer. Cao et al. (2020) focus on the contextual token embeddings layer, and claim that good multilingual spaces should have well-aligned contextual token embeddings; they describe an approach to quantify this alignment given parallel sentence corpora, as well as an optimisation approach to improve alignment.

2.3.2 How do they emerge?

A small (albeit very thorough) set of papers has addressed the problem of isolating the factors that lead to the creation of an efficient multilingual space within large language models, either by training full models from scratch, or toy models – on, for instance, synthetic data.

One such work (K et al., 2020) provided a very comprehensive study of the multilingual capabilities of a bilingual BERT model; the authors train bilingual models on the English-Spanish, English-Russian and English-Hindi language pairs. The authors first examine how corpus similarities could lead to multilingualism. They corroborate Wu et al.'s observation that shared anchors do not have a large effect on multilingual capabilities, by creating 'fake' variants of their English corpus by shifting BPE indices up by a huge constant, such that there was no overlap between BPE vocabularies for both languages. They also reject unigram frequency as a potential anchor, showing that word order similarity has a much stronger effect; however, in their experiments, permuting training corpora still resulted in better-than-random performance, implying that it could not be the only factor. Next, they examine BERT's architecture; they show that neither the total number of parameters (past a certain threshold) nor the number of heads in the attention mechanism is as useful for building multilingual spaces as the depth of the model is. Finally, they examine training objectives and show that a) the next-sentence prediction task hurts cross-lingual performance; b) adding language identification markers does not affect performance; and c) word/word-piece tokenisation works better than character-level tokenisation.

Artetxe et al. (2020) addressed earlier works pointing out that mBERT's multilingual ability could emerge due to anchor points and joint training; they design a multilingual transformer language model using an entirely different approach that strips away these potential 'clues'. Briefly, they use a two-step training process, wherein they first train a monolingual masked language model on English, and then train *only the token embeddings* on another monolingual corpus; they freeze the transformer body, swap out the English token embeddings with those of the new language, and train only the embeddings layer. When they fine-tune their model, they first use (frozen) English token embeddings, fine-tune the rest of the transformer, and swap out the English token embeddings during evaluation. They evaluate their models on a range of multilingual tasks. Critically, they experiment with shared and disjoint vocabularies for their two languages, and show that a shared vocabulary is *not* necessary for multilingualism. They find that, instead, effective vocabulary size per language was an important factor, and that their approach works best with large disjoint vocabularies, that effectively guarantees that each language has a large number of vocabulary slots reserved for it.

Dufter and Schütze (2020) continue along similar lines: the authors analyse what elements are necessary for multilingualism. They train smaller, stripped-down bilingual BERT models, with smaller vocabulary capacities, on bilingual corpora: in their more naive setup, these languages were English and K et al.'s

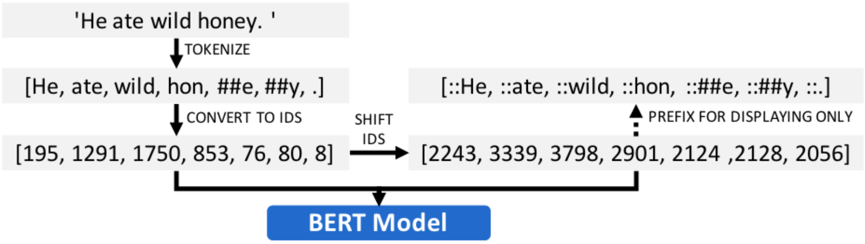


Figure 2.3: An illustration of the English/fake-English training setup; image taken from Dufter and Schütze (2020)

BPE-shifted fake-English. They evaluate these models on word translation and sentence retrieval – these are deterministic tasks given their training setup, as both words and sentences can be translated to their BPE-shifted equivalents. Dufter and Schütze corroborate prior observations about word-order similarity being important for multilingualism; they also show that *position embeddings* (discussed in greater detail in §4.3) are critical for building multilingual spaces. In Paper III, we build on this setup and attempt to describe the role that language and corpus choice plays on the quality of bootstrapped multilingual spaces.

Relevance

The specific relevance of our work to this section is our analyses of how differences in the training/fine-tuning corpora reflect on the multilingual performance (for some aspect of 'multilingual') of language models. In Lauscher et al. (2020), we addressed the limitations of zero-shot learning: we showed that zero-shot transfer performance tends to be poor when the source and target languages are not closely related. We measured the impact of *few-shot* learning, which differs from zero-shot learning in that the model is fine-tuned on (very) few additional annotated instances in the target language, and we showed that for a range of tasks and languages, it often makes sense to invest time/money into annotating very small corpora, as the benefits as far as increased performance is concerned are often large.

In Ravishankar et al. (2021b) we shifted focus from the more popular BERT to ELMo. The original mBERT implementation used exponentially smoothed weighting to undersample larger corpora (see §2.2.2); we experimented with a range of αs in an attempt to quantify the effect that the relative sizes of different language corpora has on the quality of a multilingual model. As part of this work, therefore, we also released several multilingual ELMo variants into the public domain.

Finally, in Ravishankar and Nivre (2022), we attempted to combine prior work on determining precisely what language factors contributed to multilingualism, with work describing the inductive biases of language models towards specific languages. Thus, we experimented with Dufter and Schütze's setup, yet started with different base languages and corpora, to determine what linguistic factors might contribute to multilingual space construction. We showed that corpus effects – rather than language-specific linguistic factors – tended to influence the quality of the multilingual space.

Chapter 3

Interpretability via probing

The recent paradigm changes in NLP – first moving from human-defined feature vectors to dense features, and then eliminating complex architectures entirely, replacing them with deep neural models of language – has led to the realisation that we do not fully understand how or why these models work. Being able to interpret or diagnose model decisions is of critical importance to a substantial chunk of real-world model usage contexts; this led to the proliferation of papers analysing what came to be known as *model interpretability*. An early approach to doing this was via *probing*, which broadly refers to a class of methods that use datasets annotated to be linguistically informative, and fitting (for instance) linear classifiers on top of deep language models to these datasets. In some sense, architecturally speaking, this is not entirely different to transfer learning; the difference is that while transfer learning aims to *maximise* downstream performance under transfer, probing aims to use the simplest possible model to *extract* information from the base language model. This distinction between classifiers for actually solving a task, and for evaluating the extent to which a task is solvable is critical to cast probing in the right light.

3.1 Probing techniques

Research into probing has generally proceeded along two parallel trains of thought: the first addressing the design of the probes themselves, and the second addressing what these probes can tell us about language models. While 'default' probes mostly involved fitting linear classifiers on top of language model representations, more complex probes – for instance, to extract structured information – have also been proposed; we describe a brief history of probing techniques in this section.

3.1.1 Probing via classification

Formally, (adopting notation from Belinkov (2022)), we denote a deep neural model by $m : x \mapsto \hat{y}$, where the model m maps input x to output $\hat{y}$. A probing classifier (informally, a probe) can be denoted by $p : m_l(x) \mapsto \hat{z}$, which maps the representations that a model generates at layer l to some output $\hat{z}$, which tends to be some feature of interest. This is very similar to classification as a task: indeed, probes had also been referred to as *diagnostic classifiers* (Hupkes et al., 2018) in their early days. Early works in this direction involved attempting to extract information from dense word representations; for instance, Gupta et al. (2015) attempted to probe word representations[1] for factual attributes of (word) referents, by training logistic regression models to predict FreeBase attributes (eg. population) of referents to countries or cities. Similar methods were used to probe word representations generated in context, as in Belinkov et al. (2017), where classifiers were trained to extract morphological and part-of-speech information from neural machine translation models. Adi et al. (2017) experiment with a range of sentence embedding methods (including averaged word embeddings) and probe them for features such as sentence length, or whether or not a word (given its representations) belonged in the sentence; they show (among other results) that, for instance, while LSTM encoders were significantly better than CBOW encoders for storing sentence length, CBOW models surprisingly showed length prediction scores far above the baseline.

Conneau et al. (2018a) present a dataset of 'probing tasks' for English; these included common linguistic tests, such as the depth of a sentence's parse tree, or the tense of a sentence (given the main verb). They experiment with a range of sentence encoding methods and evaluate the extent to which each such linguistic property was extractable from the representations that it generated. We base Papers IV and V on their work, and create multilingual versions of their dataset; in Paper IV we evaluate sentence embeddings generated as in Conneau et al. (2018b), and in Paper V we analyse deep pretrained multilingual models.

3.1.2 Moving beyond classification

Word- or sentence-level probing is fairly limited, in that the most one can probe for is linguistic properties specific to words or sentences. Tenney et al. (2019b) define *edge probing*, where they describe an approach to probing spans of tokens: in addition to training a probe, they further train parameterised pooling functions to compose representations for tokens in a given span. Using their formalism: if a model's input x consists of tokens $x_1, x_2, ...$, we train a pooling function $f : [m_l(x_i), ..., m_l(x_j)] \mapsto z$, and instead redefine our probe as $p : z \mapsto \hat{y}$.

The authors applied this method to probe representations from ELMo, GPT, and BERT for knowledge pertaining to a range of tasks, ranging from part-of-speech tagging to coreference resolution. In a follow-up paper focusing on BERT representations (Tenney et al., 2019a), they probe BERT's representation layer by layer, and show that information pertaining to a variety of tasks tends to resemble the 'classical' NLP pipeline, where outputs generated from models trained on specific tasks could be used as inputs for models for higher-level tasks. They show a similar localisation of task information within BERT models: in their experiments, BERT's lower layers contained more extractable information pertaining to lower-level tags like part-of-speech tags, while higher-level semantic tasks, such as relation extraction, were best extracted from BERT's higher layers. An earlier work (Peters et al., 2018a) applied a similar probe to

[1] While early works do not frame their research questions in these terms, to *probe X for Y* has become the standard way to describe such questions.

ELMo layers, with similar results: lower layers were shown to specialise in local syntactic relationships, while higher ones specialise in more complex semantic tasks.

An alternative approach that emerged in response to classifier-based probing, particularly probing for evidence of linguistic structure, was to use *structural probes*. Hewitt and Manning (2019) define a probe as a set of parameters B; they use gradient descent to ensure that the probe-transformed distances between representations at i and j are as close to the distance between the two tokens as possible, i.e., given representation $\mathbf{h}$ for word w, for a single sentence, they approximate:

$$\min_B \frac{1}{|s|^2} |d_T(w_i, w_j) - d_B(\mathbf{h}_i, \mathbf{h}_j)^2|$$

where the parameterised distance $d_B(\mathbf{h}_i, \mathbf{h}_j) = (B(\mathbf{h}_i, \mathbf{h}_j))^\top (B(\mathbf{h}_i, \mathbf{h}_j))$. We build on precisely this probe in Paper VI to evaluate the extent to which models like BERT show preferences for specific syntactic annotation schemata. More specifically, we compare the Universal Dependencies annotation schema to SUD (surface syntactic UD) annotations, a schema that differs in certain aspects (such as that it does not privilege content words as syntactic heads).

Parameterised probes, while conceptually simple and extensible, came with a host of issues (that we address in §3.2). This led to the proliferation of unsupervised probing techniques, that could, for instance, rely on an examination of the model's internal representations. Note that this is where the distinction between generic model intepretability and *probing* as a form of interpretability started to become a bit fuzzy; conventionally, probing was a term used to describe forms of interpretability that involved extracting information from *representations* generated by models. Several unsupervised probing techniques fit into this description: for instance, perturbed masking (Wu et al., 2020b), where the authors use the impact a word w_j has on the prediction of another word w_i – as in $f(w_i, w_j)\ \forall i, j$ – and used the resulting *impact matrix* to (for instance) extract syntactic trees.

Another such approach involved the use of comparison-based methods like *representation similarity analysis* (RSA) (Kriegeskorte et al., 2008), which uses correlation metrics to measure 'distances between distances', so to speak (see Figure 3.1). Given a set of sentence encoders, RSA involves first constructing, for each feature encoder $f_m \in \mathbf{F}$, a first-order (dis)similarity matrix for features generated for each (say) sentence in a corpus, i.e. a matrix M such that $\mathrm{M}^m_{ij} = \rho(f^m(w_i), f^m(w_j))$. Note that a feature encoder can be anything that returns some sort of feature vector (such as a language model representation). Next, a second-order similarity matrix N can be computed in order to compare two encoders f_m and f_n; the higher the correlation between M^m and M^n, the more the two representation methods agree on their representational geometry for a set of sentences. This matrix can be of interpretable value if one such encoder was some sort of well-defined metric, such as tree kernel metrics (Chrupała and Alishahi, 2019); alternatively, it can reflect measurements

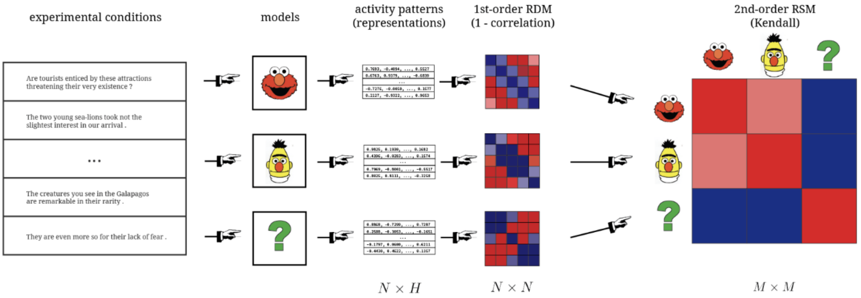

experimental conditions
models
activity patterns (representations)
1st-order RDM (1 - correlation)
2nd-order RSM (Kendall)
Are tourists enticed by these attractions threatening their very existence ?
The two young sea-lions took not the slightest interest in our arrival .
...
The creatures you see in the Galapagos are remarkable in their rarity .
They are even more so for their lack of fear .
?
$N \times H$
$N \times N$
$M \times M$

Figure 3.1: Second-order comparisons using representation similarity analysis; image taken from Abdou et al. (2019).

of the agreement between two LM encoders, which could further be compared with real-world metrics coming from, for instance, eye-tracking data (Abdou et al., 2019).

Yet another comparison-based approach to probing was the use of analytical methods like *singular vector canonical correlation analysis* (SVCCA) (Raghu et al., 2017); Saphra and Lopez (2019) use SVCCA to compare representations generated by a language model over the course of training to representations generated by a fully-trained POS tagger. They first reduce representation dimensionality (singular value decomposition/SVD), and then project the representations to be compared onto a shared subspace (canonical correlation analysis/CCA). Kornblith et al. (2019) propose using *centred kernel alignment* (CKA), an improved metric that is robust to different random initialisations of a network.

Other works that discussed unparameterised probing tended to involve an inspection of the model's intermediate representations; as far as transformers are concerned, these typically tended to be the parameters of the attention mechanism. I defer a more detailed discussion of this class of methods to Chapter 4.

3.2 Findings and criticism

While there have been a fair number of interesting conclusions and findings drawn from probing tasks, these tasks also faced criticism from a variety of lenses, many arguing that probes represent an insufficient insight into model behaviour. We summarise both insights and critiques here; we refer the reader to Belinkov (2022) for a detailed survey on probes, including criticism thereof.

3.2.1 What have we learnt about language models?

Somewhat parallel to research into probing *methods*, there had been a considerable amount of research that involved using probes for their intended purpose:

to determine what precisely language models had learnt.

Syntax Early influential works discussing how syntax is represented within language models include, for instance, the structural probe we discussed in the previous section (Hewitt and Manning, 2019). Related works include, for instance, Chi et al. (2020); the authors apply structural probes to mBERT, to determine whether syntactic subspaces were shared across languages. They show that this is indeed the case: structural probes could be trained on one language and transferred to another. Further, these dependencies cluster cross-lingually by relation, even when not explicitly annotated.

Morphology Edmiston (2020) probe BERT models for Universal Dependencies morphological information; they show that in general, morphological information is extractable from the model; however, unlike syntactic role, this information would need supervision to reflect in the representations. They also show that case syncretism is significantly opaque for the models (unlike for humans), and backed up earlier claims that early-middle layers prioritise morphology, particularly for more morphologically complex Indo-European languages like German and Russian. Hofmann et al. (2020) analyse BERT along a lens of *derivational* morphology, and show that BERT is capable of predicting the appropriate derivational suffix in context (for English), albeit when provided with morphologically accurate segmentation.

Semantics Wiedemann et al. (2019) analyse contextualised word representations produced by BERT and ELMo; they cluster them to determine the extent to which different word senses are separable into separate clusters, showing very effective performance. Vulić et al. (2020b) evaluate word representations from monolingual and multilingual BERT models, along a series of metrics (lexical similarity; word analogy; bilingual lexicon induction; cross-linguistic information retrieval). They find that monolingual models contained far more language-specific lexical semantic information than multilingual models, even for languages like English.

Other, broader probing suites that attempted to evaluate BERT's syntactic strengths included Jawahar et al. (2019): the authors run a variety of analytical tests, including probes for Conneau et al.'s suite of sentence metrics, layer by layer. Liu et al. (2019a) probe token/token-pair representations from a range of language models, for their performance on a large suite of downstream tasks, ranging from part-of-speech tagging to grammatical error detection. Both these works show, contemporaneous with Tenney et al. (2019a), that BERT's layers tend to resemble the classical NLP pipeline.

3.2.2 Where does probing fail?

An issue with probing is the philosophy behind the selection of an adequate probe. There is general agreement that probes ought to be 'simple', in order to ensure that the probe itself is not capable of learning information that affects performance on the downstream task (Hupkes et al., 2018; Liu et al., 2019a); however, quantifying the simplicity of a probe is difficult, despite being essential for qualifying precisely what a probe's results on a probing task mean. An option is to use randomly initialised models as baselines, as in Zhang and Bowman (2019); they find, however, that these random baselines did not differ from pre-trained models very substantially. Further, the gap narrows the longer the probe is trained, emphasising that even simple probes can learn task-specific information. Thus, to quote Hewitt and Liang (2019, p. 2733), "as long as a representation is a lossless encoding, a sufficiently expressive probe with enough training data can learn *any* task on top of it".

Several solutions to this conundrum were proposed. Pimentel et al. (2020b) describe *Pareto probing*, wherein they propose using Pareto optimal probes – i.e., probes for which no competing probe with higher accuracy yet lower probe complexity exists. Hewitt and Liang (2019) propose using *control tasks* to ensure easier probe contextualisation; they assign random labels to tokens, drawn from the original label distribution, and defined selectivity as the difference between model performance on an actual probing task and on a control task. They caution against using MLP probes, as they have poor (i.e. low) selectivity on part-of-speech tagging and dependency parsing. This method, however, had the issue of turning probing into an architecture search problem, wherein one would have to find task-specific probes that both showed poor performance on control tasks, yet were functional *as probes*. Voita and Titov (2020) suggest instead using *minimum description length* (MDL) as a measure of model performance on a probing task, rather than probe accuracy; thus, as a proxy for the extractability of linguistic information pertaining to some probing task, they describe a method that involves examining the number of bits required to transmit the task data, as well as the compression model itself.

Note that other works, such as Pimentel et al. (2020a), take issue with the generally accepted idea that probes ought to be as simple as possible; they presented the question regarding whether a probing classifier is merely *extracting* information from an encoder or whether it has learnt the task itself as a false dichotomy, and instead proposed using an information theoretic framework: they use, for a set of tags T given representations R, some control function $\mathbf{c}$ such that $I(T;R) \geq I(T;\mathbf{c}(R))$, where I represents mutual information. They experiment with control functions that would return fastText embeddings, or one-hot embeddings, and use this mutual information as a baseline, rather than using a control task[2].

Relevance

Ultimately, much of the progress in the field of probing took place parallel to our research. As such, two of our papers relevant to this domain involved

[2] Note that both approaches have been shown to be theoretically equivalent (Zhu and Rudzicz, 2020).

concretely running probes based on multilingual datasets that we generated, inspired by Conneau et al. (2018a). First, in Ravishankar et al. (2019c), we generated datasets for probing tasks in 5 languages (including English); we then analysed multilingual sentence encoders, learnt via transfer learning, applied to English encoders. Interestingly, we found that several linguistic properties – particularly sentence length – tend to be *more* extractable from these transferred representations. In a follow-up work (Ravishankar et al., 2019a) – given the backdrop of the dominance of deep multilingual language models – we first extended our dataset to seven languages (including English), and then probed ELMo, BERT and XLM models for the same properties. We then analysed our results through a lens of task, language, and encoder.

Finally, in Kulmizev et al. (2020b), we applied Hewitt and Manning's structural probe (2019) to quantify how a language model's syntactic extractability differs when given a different syntactic formalism, i.e. surface Universal Dependencies (SUD). We found that syntactic trees tend to be more extractable using a structural probe where trees are shallower; i.e., we found a preference for UD over SUD in languages that tend to have a higher proportion of function words, and thus deeper SUD trees.

Ultimately, given the robustness of the debate surrounding the usefulness of probing, and the numerous caveats in taking probing results too literally, as well as the disagreement concerning what an adequate probe should look like, we began to explore alternative interpretability methods: for instance, methods that involved actively inspecting model architectures. We describe research in this direction, along with a primer on the architecture of the transformer model, in the next chapter.

Chapter 4

Interpreting transformer language models

Early neural NLP involved much discourse about and innovation in the specific *methods* used to process language. Several such methods were inspired by our intuitions regarding human processing of language, such as *recurrent neural networks* (RNNs) or *convolutional neural networks* (CNNs). RNNs were an early class of deep neural architecture for language processing, wherein tokens would be processed sequentially; while the internal architecture of the model depended on the specific class being used. However, a common element to each was that that each token w_t, when processed by a recurrent cell h_t, would output an output representation y_t, as well a context vector c_t; the next recurrent cell, h_{t+1} would rely on both w_{t+1} and c_t to output y_{t+1} and c_{t+1}; this process would continue until the entire input sentence was consumed. We avoid more detailed descriptions of RNNs, as their internal functionining is mostly irrelevant to this thesis.

Transformers emerged around 2017 as a highly parallelisable alternative to RNNs, which dominated much NLP research at the time. A critical component of transformers was the *attention mechanism*, which originated in the context of neural machine translation, and saw much use in conjunction with RNNs or CNNs. We therefore begin this chapter with a description of the attention mechanism, and the role it plays in the transformer. We then address (specifically relevant to our research on multilingual language models) the various components that make up the transformer, and focus on a select few, describing their role in interpreting model behaviour, and their relevance to our research.

4.1 Transformers

The transformer's fundamental building block is the attention mechanism. Attention, initially described by Bahdanau et al. (2015), was partially motivated by its easy visualisability. This paper was written against a backdrop of discussions pertaining to the interpretability of deep-learning based machine learning solutions, and their applicability in the real world, given the black-box assumption. Attention, initially described in the context of machine translation, was meant to be an easily interpretable mechanism that would, unlike post-hoc explanatory methods, *inherently* describe the reasoning process of a model; for instance, consider Figure 4.1, which is a visualisation of the magnitude of the attention paid by tokens in a source sentence to a target sentence, in the context of machine translation.

4.1.1 The maths behind attention

In its simplest form, attention involves building *context vectors* by weighting inputs. Consider, for instance, machine translation, with RNNs as the base architecture, with c being the last recurrent state, obtained after processing the entire sentence – essentially, a vector representation of the entire input sentence. Conventionally, this context vector c could be used to generate some sequential outputs, such the translation of the source sentence, $\{s_0, s_1, ..., s_N\}$. This is *functional*, in that the generation of a particular output token is conditioned on both the previously generated outputs (enabling *fluency*), and a vector representing the input (enabling *adequacy*).

The attention mechanism was motivated by the observation that a one-size-fits-all context vector c made little sense; each output token should, ideally, be conditioned on a different 'view' on the input. The attention mechanism was therefore a proposed replacement of this vector c, with output-specific *weighted sums* of input context vectors. Thus, for an output token at output step i, attention would generate a context vector c_i; this c_i could be calculated based on the relevances of every input hidden state $\{h_0, h_1, ..., h_N\}$, to the output being generated at step i. As a weighted sum of input embeddings, therefore, $c_i = \sum_{j=1}^{N} \alpha_{ij} h_j$. In this equation, α is a matrix of weights conventionally referred to as an *attention weight(s) matrix*; each element of α_{ij} encodes the relevance of input token j to output token i[1]. The discourse surrounding the alleged interpretability of attention stems from precisely this definition of attention – if attention was, at its core, a set of importance weights, it was seen as plausible that these weights could have some explanatory value when it came to describing model decisions.

Calculating α_{ij}, as defined by Bahdanau et al., is simple: α is effectively a probability distribution applied to some alignment model e, i.e. $\alpha_{ij} = \exp{(e_{ij})} / \sum_{k=1}^{N} \exp{(e_{ik})}$ – which is essentially a softmax. Bahdanau et al. propose simply using some alignment model a, i.e. $e_{ij} = a(s_{i_1}, h_j)$. In their paper, they use a single-layer feedforward network with a non-linear activation function; Luong et al. (2015) contrast and evaluate different kinds of alignment models.

Attention was rapidly adopted as an additional component – often built on top of recurrent neural networks – that could push state-of-the-art results for a range of tasks. Tasks such as machine translation (Luong et al., 2015; Sennrich et al., 2016), summarisation (Rush et al., 2015), document classification (Yang et al., 2016), tasks involving sentence pairs Seo et al. (2017) and Wang et al. (2018) saw significant performance improvements with the addition of attentive components.

[1] In common parlance, this is referred to as the "attention paid by token *i* to token *j*" – note that the output token is the token doing the 'paying' of attention.

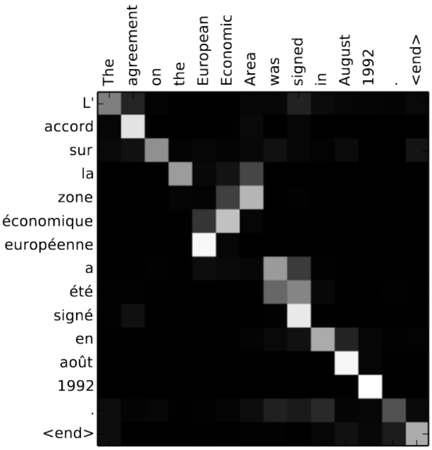


Figure 4.1: An easily interpretable attention matrix, mapping French tokens to their English counterparts, taken from Bahdanau et al. (2015); the colour of each square denotes the strength of the attention paid.

4.1.2 From cross- to self-attention

The fundamental component of the transformer architecture (Vaswani et al., 2017b) was *self-attention*, wherein word representations were derived based on tokens in an input sentence paying attention *to each other* – cf. the earlier form of attention we described, now also called *cross*-attention, where sentences would pay attention to outputs being generated, or to other paired inputs. Conceptually, in a traditional input/output scenario, calculating the attention paid by output token i to input token j would involve calculating the *relevance* of j to i. In the context of self-attention, the relevances of *input* token i to every *input* token j could be used to generate representations for i. Thus, using our earlier notation, e_{ij} would equal $a(h_i, h_j)$, for some pair of input representations with indices (i, j).

Vaswani et al. proposed first using the vector dot product as the alignment function, i.e. $a(h_i, h_j) = h_i \cdot h_j \, \forall i, j = \text{HH}$. Having calculated alignments, they then apply three separate linear transformations to obtain three different 'views' on the token embeddings: a *query* representation, representing some 'source' token, a *key* representation, representing a 'target' for calculating attention weights, and a *value* representation, which was multiplied by its corresponding attention weight; these weighted value representations were summed to obtain a token word representation. Conceptually, this meant substituting the word embeddings matrix H with key, query, and value different matrices, i.e. K, Q and V. The first two would learn to represent tokens in their roles as tokens paying attention, and as tokens being paid attention to; the last would learn to form token representations that could be weighted and combined to form full contextual representations.

Thus, mathematically, the attention weights matrix $\alpha = \mathrm{softmax}\left(\mathrm{QK}/\sqrt{d_k}\right)$, where d_k was a scaling factor equal to the embedding dimensionality; the final representation matrix $\mathrm{Z} = \alpha\mathrm{V} = \mathrm{softmax}\left(\mathrm{QK}/\sqrt{d_k}\right)\mathrm{V}$.

4.1.3 Putting everything together

A few additional innovations that made up the transformer architecture are worth mentioning. The first of these was the extension of self-attention to *multi-headed self-attention*, which effectively involves calculating multiple attention weight matrices using different sets ('heads') of parameters, then concatenating their representations, in the hopes that each attention head would learn a different linguistic phenomenon. The next was that of *position embeddings*; given that the transformer consumes all input tokens in parallel, there is no sequential ordering of data, which reduces the transformer to a bag-of-words model. Position embeddings were a way to avoid this; we discuss them in greater detail in §4.3.

The transformer, as described in the context of sequence-to-sequence tasks (like machine translation) consists of two 'halves': an encoder and a decoder. Figure 4.2 is a diagram of the transformer; the inputs correspond to the encoder, and the outputs to the decoder. The encoder is the non-autoregressive half: the entire input sequence can be modelled simultaneously, effectively utilising the parallelisability of the transformer. The decoder generates an output sequence token by token; each generated token at a given time step is consumed, in addition to cross-attention with the encoder, to generate the output token at the next time step. Both halves consist of stacked attention layers, each of which consists of multi-head self-attention, LayerNorms (Ba et al., 2016), and feed-forward networks. In the context of language modelling, both encoders and decoders saw widespread use as independent models: the former in the BERT/RoBERTa class of models, and the latter in the GPT family of language models.

4.2 Attention as explanation

As mentioned at the start of this chapter, part of the motivation for attention, beyond its parallelisability, was its alleged inherent interpretability. This motivation became somewhat murkier with the introduction of the transformer, though visualisation mechanisms continued to exist (Vig and Belinkov, 2019). For one, self-attention was not as clearly intuitive as cross-attention; next, the addition of layers and heads made it unclear how reliable naive visualisations of attention weights could be. This naturally sparked considerable discussion that we shall attempt to summarise in this section, and also examine in light of our contribution to this strand of research.

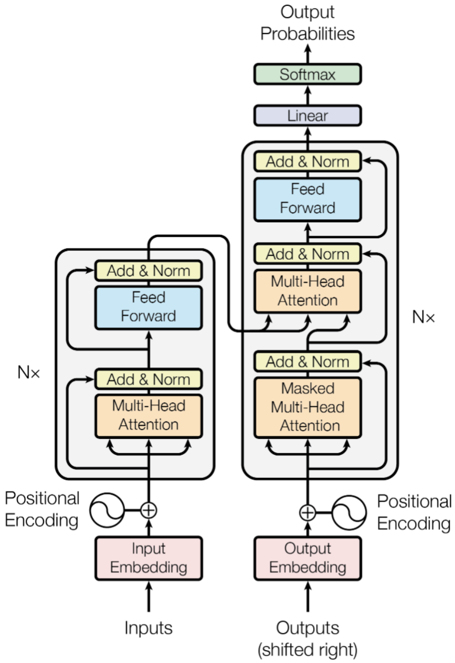


Figure 4.2: The transformer model. Image taken from Vaswani et al. (2017b)

4.2.1 The state of the debate

The debate on the role of attention as explanation was sparked by Jain and Wallace's work (2019), titled *Attention is not Explanation*. The authors run a series of experiments: first, they compare correlations between learned attention weights on an LSTM-based classifier, and a) gradient-based measures of feature importance, or b) differences in model output, induced by leaving out single input tokens; they show that these correlations tended to be rather weak. Next, they experiment with 'counterfactual' attention, either by permuting attention, or by actively inducing attention distributions that differ from learned attention, showing only a modest effect on output labels. Based on these experiments, they claim that while attention is undoubtedly useful, it is unclear whether or not it had any explicitly interpretative value, given that apparently arbitrary attention weights also led to reasonable downstream performance.

A response to their work, titled *Attention is not* not *Explanation* (Wiegreffe and Pinter, 2019), addressed some of their claims. The authors of this latter work first show that uniformly initialised attention weights perform just as well as learned attention on some tasks, implying that there exist tasks for which attention is entirely *unnecessary*, and not necessarily just obscure. They then replace the LSTM from the classifier with a single MLP – implying shared parameters for each token – and then experiment with different mechanisms for weighting the

MLP output for each token. They find that using the learned attention weights from the LSTM-based model outperforms both the MLP weights learnt during training, as well as their previous uniform-attention LSTM model. From this, the authors conclude that attention weights cannot be truly arbitrary, or model-dependent, given that MLP models that lack access to contextual information found them useful – they claim that while attention may not be *the* explanation for a model's decisions, it is clearly *an* explanation.

Serrano and Smith (2019) added to this debate by lowering the bar for what constituted interpretability. They focus on specific attention *layers*, zeroing out a single high-valued attention weight, renormalising, and analysing shifts in the output label distribution. They find that while attention does somewhat correspond to importance, in that higher attention weights tend to have a larger impact on the model's decisions, their experiments show that there are numerous cases where the impact of these high-valued attentions is minimal. They thus take a more pessimistic view than Wiegreffe and Pinter (2019), claiming that attention often fails to explain model decisions, when viewed as a feature ranking.

4.2.2 Observations from the sidelines

Parallel to this debate, Bastings and Filippova (2020) ask whether the debate around attention being explanation was meaningful at all, and whether we should even care, given the existence of *saliency methods* as established interpretability methods. Saliency methods are, briefly, a family of interpretability mechanisms often used in the context of computer vision, wherein regions of an image are highlighted based on their relevance to model decisions; in the context of NLP, this selection and highlighting is often carried out at a word level[2]. Similarly, Jacovi and Goldberg (2020) argue that there was no consistent, formal definition of what constituted a *faithful* explanation in the literature, and called for methods to be defined as *sufficiently* faithful – that is to say, faithful in the context of some models and tasks, or certain parts of the input space.

Other works imposed constraints and caveats on what elements of attention ought to be considered when discussing the interpretability of attention. Brunner et al. (2020) discuss the issue of *identifiability* within attention mechanisms, i.e. the degree to which attention weights at higher layers truly derived from their corresponding tokens, or, conversely the lack of a true correspondence between a token at index i and its corresponding attention weight at layer L (α_i^L). The authors describe a method to quantify this degree of information mixing that they termed *hidden token attribution* (HTA), and suggest that these caveats ought to be taken into account when discussing the interpretability of attention. Thus, they proposed examining what they termed *effective attention* – a lens that would involve removing attention weight components that did not affect model predictions. They re-analyse attention patterns in masked

[2] Note that it is not entirely clear that saliency methods themselves provide consistent explanations for model behaviour (Krishna et al., 2022).

language modelling (Clark et al., 2019) in light of this knowledge, and find that attention weights corresponding to [CLS] and [SEP] rapidly collapsed. Sun and Marasović (2021) analyse effective attention, and found that it tends to view linguistic features as more important; however, they fail to replicate Brunner et al.'s results on attention collapse for [CLS] tokens given a wider variety of downstream tasks.

Building on these observations, Pascual et al. (2021) apply HTA and demonstrate that attention patterns that take token mixing into account differ from vanilla attention weights. Abnar and Zuidema (2020) also take token mixing into account, and describe *attention rollout* and *attention flow*, two mechanisms to address this lack of token identifiability at higher transformer layers. They find that applying either post-hoc method yields better correlations with importance scores. Relevant also was Kobayashi et al.'s work (2020); they argue that attention weights could not be observed in isolation, as they are used to weight specific value vectors – the representation for token i, i.e. y_i, is defined as the sum of the products of the value-transformed input $v(x_i)$, and the attention paid by every other token j to i. i.e. $y_i \propto \sum_{j=1}^{n} \alpha_{i,j} v(x_j)$, where α is the attention weights matrix. The authors reason that while a particular attention weight $\alpha_{i,j}$ might be exceptionally large or small, that said very little about the weight of the product $\alpha_{i,j} v(x_j)$, which is what the network used downstream; thus, they propose analysing *weighted transformed vector norms*, i.e. $||\alpha f(x)||$ instead of raw weights; they show, in the context of neural machine translation, that word alignments extracted using this approach, rather than using vanilla attention weights, tend to have significantly lower alignment error rates.

Specifically relevant to our work is the interaction between syntactic structure and attention: i.e. the question of quantifying the extent to which syntactic structure can be encoded within attention weights. Relevant to this was Raganato, Tiedemann, et al. (2018), where the authors attempt to extract syntactic structure from attention weights; their approach to this was to extract the *maximum spanning tree* (MST) through attention weight matrices (Chu, 1965). Clark et al. (2019) instead examind attention weights, and corroborate Raganato, Tiedemann, et al.'s results: *some* degree of syntactic structure was encoded within attention weights, but not a lot even when compared to naive baselines, like deterministic right-branching trees.

In Paper VII, we show that their observations hold true across a typologically diverse set of languages; we also show that attention very rapidly converges to patterns that resemble syntactic structure, *if* models are fine-tuned on explicitly annotated syntax.

4.3 Moving beyond attention

While attention has often been prioritised in the literature as a critical lens through which model decisions can be interpreted, some of the work summarised in this thesis has focused on other components; specifically on *position embeddings*. In this section, we describe the motivation behind the use of position embeddings in transformers, and how specific embedding methods influence model behaviour.

4.3.1 Position in transformers

In the context of masked language modelling, a critical apparent weakness of transformer encoders is their absence of word order information. While the absence of sequentiality is a positive from the perspective of parallel computing, it effectively reduces language models to overparameterised bag-of-words models; i.e. default transformer encoders are *order invariant*: their output does not change with reorderings in their input. A proposed alternative to autoregressive language modelling has, therefore, been the use of *position embeddings*. Originally – as in Vaswani et al. (2017b) – position embeddings were defined as fixed sinusoidal waveforms, with each dimension of an embedding vector representing a waveform with a different frequency and phase. Thus, position embedding p could be defined as

$$p_{k,2i} = \sin(k/10000^{2i/d}) \tag{4.1}$$

$$p_{k,2i+1} = \cos(k/10000^{2i/d}) \tag{4.2}$$

for embedding dimension i and token k, given a d-dimensional output representation.

Naturally, this 'naive' embedding method led to alternative embedding mechanisms being proposed. BERT (Devlin et al., 2019), for instance, as well as BERT-based models like RoBERTa (Liu et al., 2019b), use learnable sinusoidal embeddings: each token in each sentence would receive a unique position embedding that derived from its position k in that sentence; the parameters for this embedding model would be learnt during model training/fine-tuning.

4.3.2 Embedding methods can be non-obvious

The position embedding methods described above – embeddings that derive from the absolute position of a particular token – are dubbed *absolute position embeddings* (APE). Ordinarily, these positional representations were fed to the network by adding them to their corresponding (non-contextual) token representations before their sum is processed downstream. This is, however, not necessarily the only way to inject positional information into a transformer. Other approaches to encoding positional information generally tend to involve directly modifying attention weights, rather than token representations. Often, these methods – owing to the two-dimensionality of attention matrices – take into account the relative offset between two tokens k_i and k_j (Huang et al., 2020; Shaw et al., 2018), rather than their absolute positions (*relative position embeddings* (RPEs)).

Numerous other extensions to position injection have been proposed, and this domain continues to be an active field of research. Ke et al. (2020)

Absolute	$((w_i + p_i)W^Q)((w_j + p_j)W^K)^\top$	Devlin et al. (2019)
TUPE	$(w_i^l W^{Q,l})(w_j^l W^{K,l})^\top + (p_i U^Q)(p_j U^K)^\top$	Ke et al. (2020)
Relative(k)	$(w_i W^Q)(w_j W^K + a_{ij})^\top$	Shaw et al. (2018)
Relative(k/q)	$(w_i W^Q + a_{ij})(w_j W^K + a_{ij})^\top$	Huang et al. (2020)
T5	$(w_i W^Q)(i_j W^K)^\top + b_{ij}$	Raffel et al. (2020)

Table 4.1: A brief overview of (some) popular position embedding methods.

recommend applying separate affine transformations to position encodings and token encodings prior to adding them; they show that the naive approach to generating an attention weights matrix α, given positional information p and token information w for tokens i and j, i.e. $\alpha_{ij} = ((w_i + p_i)W^Q)((w_j + p_j)W^K)^\top$ – where Q and K are query and key parameters respectively – tends to be a wasteful operation. The expansion of this equation would lead to product terms such as $(p_i W^Q)(w_j W^K)^\top$; the authors claim that these terms are noise, as they saw a) position-token correlations as mostly meaningless, and b) positions and tokens as being different enough that sharing weights between them made little sense. He et al. (2021) take a slightly contradictory approach, and propose *adding* token-position correlations for RPEs, that lacked them out-of-the-box; they claim that the relative position between two tokens was meaningful when attempting to fully model words[3]. Wang et al. (2019) propose taking structure into account, and replace linearly assigned attention weights with weights derived from syntactic parse trees. T5 (Raffel et al., 2020), a large-scale language model built on sequence-to-sequence modelling, uses a fairly 'simple' relative positional bias, that involves merely adding a bias term $b_{ij} = W(j - i)$ to every element in α_{ij}; this differes from Shaw et al.'s encoding methods, where they add these bias terms to the transformed key and/or query vectors before multiplying them.

Some of these methods are (mathematically) summarised in Table 4.1.

Dufter et al. (2022) present a concise survey of position embedding methods in the literature. They further describe position embeddings on the basis of whether they constituted sequences, trees, or graphs, while retaining the original characterisation of position embeddings as APEs and RPEs. Wang et al. (2021) present a more theoretical survey, wherein the authors describe theoretically motivated characteristics that they claimed position embeddings ought to exhibit, and analysed APEs and RPEs along these lines.

4.3.3 Extrapolation and composition

A key requirement from position embeddings is their ability to *extrapolate* to positions unseen during training; this was a key motivation behind the original sinusoidal embedding method, as a deterministically calculated periodic function ought to be able to extrapolate indefinitely. However, Press et al. (2021)

[3] Note that Ke et al. (2020) claim that this does not contradict their work, which focuses on *absolute* embeddings.

show that this was not the case: when experimenting with language modelling, they show that for a validation set with sentence lengths $L_{val} > L$, where L was the longest training sequence, performance for sinusoidal embeddings would rapidly drop for $L_{val} > L + 50$. Rotary position embeddings (Su et al., 2022; Wang and Komatsuzaki, 2021), where sinusoidal embeddings are multiplied by the keys and queries of every transformer layer, extrapolate much better. Contemporaneous with a lot of our research, Press et al. took inspiration from rotary position embeddings and introduced ALiBi: they do away with learned position entirely, and introduce a static bias term to the key-query product term $\mathrm{softmax}(q_i K^\top + m \cdot [-(i-1), \ldots, -2, -1, 0])$, where m is a head-specific slope term drawn from a geometric sequence. They find that their method outperforms alternative methods.

Another aspect of language modelling that is worth mentioning from the perspective of this thesis is *compositionality*, i.e. a language model's ability to compose representations for words into phrases, and phrases into sentences etc. The relationship between generalisation (extrapolation being a form of generalisation) and compositionality has been studied, and is seen as somewhat complex. Chaabouni et al. (2020) show, in an emergent language scenario, that compositional languages were easier for agents to learn; further, they show that compositionality was sufficient (albeit not necessary) for the emergence of the ability to generalise. Hupkes et al. (2020) analyse the explicit extrapolatability of composition; they analyse the compositional ability of language models on validation sets with sentences longer than training sentences, and find that while transformers outperform LSTMs and CNNs, no language model could extrapolate to indefinitely long sequences. In Paper VIII, we analyse compositionality from a multilingual perspective. Pragmatically, improved compositional abilities ought to allow for improved multilingual transfer, as the model can then better utilise its multilingual capacity (see also §2 for a discussion on the factors that enable multilingualism). We show that there exists a tradeoff between 'overengineered' position embeddings, so to speak, and multilingual transfer capabilities.

4.4 Word order in transformers

Wrapping up the architecture-driven analysis of transformers, we shift focus somewhat to discuss the influence of *word order* on language models. This is an architecturally relevant question – relevant to position embeddings, specifically, on account of the fact that position embeddings are the *only way transformer encoders can model position*[4]. Our research in this direction is, in some sense, a contribution to the (growing) body of literature addressing the extent to which transformers even need word order to solve specific tasks; we discuss what these implications say about our tasks, particularly in light of what we know about *human* processing of language.

[4] Note that this is not true for decoders: see Haviv et al. (2022).

4.4.1 Language models use word order

On the face of it, it seems uncontroversial to say that human language processing relies on the ordering of words in a sentence; this may be more or less true depending on how free word order is in a particular language. Mollica et al. (2020) examine the effects of various forms of scrambling on human sentence processing. They show that composition could take place when certain grammatical constraints were violated; they also show that a word's immediate neighbourhood is more immediately critical to composition. It is tempting to claim that this must, therefore, also be the case in neural language models – however, neural language models do *not* have the same learning process that humans do, nor are they architecturally similar to human brains.

Analysing the influence of word order on language models, Sinha et al. (2021b) show that applying permutations to sentences does not necessarily affect downstream language model performance on textual entailment tasks; they also show that humans tend to be more sensitive to the same perturbations than LMs. Clouatre et al. (2022) draw on Mollica et al.'s observations concerning local vs. global structure; they show that for a subset of NLP tasks, as long as local structure is maintained, global structure is be unimportant. They also show that conventional perturbation functions tend to mostly alter global structure, and raise the question of whether or not some NLP tasks are solveable by bag-of-words models, thereby making for poor test suites, solvable through simple heuristics that are not immediately obvious: see, for instance, Poliak et al. (2018), where the authors address precisely this problem in the context of natural language inference. Alleman et al. (2021) show similar results with a representation distortion-based probing setup; they show that BERT representations are sensitive to phrasal units, and that the model's sensitivity to perturbations is mediated by attention. Somewhat contradictory, however, was a followup paper (Sinha et al., 2021a) to the entailment paper, where the authors proceed to show that language models pre-trained on perturbed text, where sentences are scrambled at an n-gram level for $n \in \{1, 2, 3, 4\}$, continue to perform competitively on a set of downstream tasks; this is perhaps the strongest (current) evidence for language model performance under the absence of word order information.

In Paper IX, we show that language models do need word order information. We address the observations in Sinha et al. (2021a), particularly their results showing that language models pretrained without position embeddings show considerably worse performance than models trained on unigram-level scrambled text: we show that some sense of word order information exists even in naive scrambling setups (such as the adjacency of subword units), and that position embeddings can leverage this information to develop the appropriate inductive bias prior to fine-tuning. We also show that the *choice of task* has a strong effect on how well 'unordered' language models perform, backing up Clouatre et al.'s suggestion that certain tasks are substantially easier to 'solve' than others, even without word order.

4.4.2 What does this say about our benchmarks?

Sinha et al.'s work, and our response, raises another question: given that unordered language models – whether due to unigram shuffling or due to the absence of position embeddings – perform fairly well on GLUE as a benchmark suite, what does this actually say about GLUE itself? Zooming out from talking specifically about GLUE, we draw the reader's attention to this passage, quoted from Church (2017, p.476):

> I worry that the literature may be turning into a giant leaderboard. As reviewing burdens continue to become more and more onerous, reviewers are looking for easier and easier ways to discharge responsibility. Papers are being rejected for silly reasons like typos, and papers are being accepted for equally silly reasons like topping a leaderboard.
>
> Leaderboards are great, but a paper should do more than merely top a leaderboard. Leaderboards provide a useful service by helping the audience figure out how the proposed solution stacks up to the competition, but that should be just a starting point to motivate a more interesting discussion on why the proposed solution works as well as it does. Such an explanation ought to call out some novel insights that distinguish the proposed solution from the competition.

The proliferation of easy-to-game benchmarking in NLP that this discusses is a particularly salient concern when discussing word order, and, particularly, of reading too deep into model performance in the absence of word order. The many issues with drawing conclusions based on model performance on test suites have been discussed in the literature. McCoy et al. (2019) explain how model decisions (in the context of entailment tasks) could often be the product of simple heuristics that the model would learn to apply, rather than due to a deeper understanding of language; Jia and Liang (2017) show how the insertion of distracting sentences could affect model performance on question answering tasks; numerous such examples exist for a variety of tasks. In light of this debate, Ribeiro et al. (2020) describe the creation of comprehensive test suites for different NLP tasks, inspired by unit tests in software engineering, that go beyond mere accuracy – they show that state-of-the-art models (at the time) exhibited considerably reduced performance on many such metrics. Bowman and Dahl (2021) discuss four criteria that they believe NLU benchmarks ought to satisfy, namely validity, reliable annotation, statistical power and disincentives for biased models.

Contextualising what we now know, and summarising our view on the necessity of word order in Paper IX, it is clear that several clues in training corpora and architectural biases, combined with heuristically easy-to-solve tasks, could contribute to the perception that word order could be unimportant, and that we would benefit from deeper examinations in the context of tasks more complex than GLUE.

Relevance

Diving into the architecture of transformer language models and analysing models along these lines became a logical follow-up to the more extrinsic probing, over the course of our PhD. In Ravishankar et al. (2021a), we dove into the attention/explanation debate. We backed up earlier work showing that out-of-the-box, language modelling based attention may not necessarily encode (syntactic) explanations; however, we showed that attention could be made to encode syntax simply by fine-tuning on syntactically annotated data, across a wide range of typologically diverse languages. We also analysed these results to attempt to tease out which attention components actually contribute to this learning process. Critically, we showed that this shift in attention being more interpretable *does not necessarily* emerge due to changes in the key and query components that make up attention weights. In our experiments, we first froze transformer components during fine-tuning, and ran Raganato, Tiedemann, et al.'s MST-based algorithm on our attention weights. We showed that while allowing either key and query components (or both) to continue fine-tuning could lead to an improvement in UUAS, it was the *value* components that led to the most substantial changes in UUAS when fine-tuned. This also furthers Kobayashi et al. (2020)'s claim that value vectors ought to be taken into account when addressing attention as a mechanism.

In Ravishankar and Søgaard (2021), we shifted our gaze away from the attentive components of the transformer to the embeddings layer, specifically to position embeddings. We argued that while some of the recent innovation in position embedding methods clearly resulted in improvements in model capabilities on monolingual tasks, the absence of multilingual evaluation also meant that we did not fully understand how these methods impact multilingualism in a language model. We showed that the 'perfect' periodicity in sinusoidal embeddings was a desirable property for multilingualism, as it enabled better compositionality at arbitrary token offsets. We showed that more complex embedding methods learnt more complex composition functions, which we hypothesised could hurt compositional generalisability across multiple languages.

Finally, in Abdou et al. (2022), we dove into the entire debate surrounding the question of whether masked language models even need word order information. We argued that they do: in doing so, we highlighted the gap in performance in Sinha et al. (2021a) between language models trained on token-scrambled corpora, and language models trained without position embeddings. We showed that language models trained on scrambled corpora retained enough word order information that a linear model built on top of LM representations could accurately predict the distance between two tokens, as well as their order in the sentence. This information could bleed into the model through certain clues – BPE segmentation was a signal that order matters somewhat; there could be natural language correlations between word use and sentence length; having position embeddings could allow models to use the inductive bias that they provide to learn these clues given a large enough corpus. We also show that there exist NLP tasks for which these clues were insufficient, and that fully ordered transformers were critical; conversely, we showed that a large chunk of mainstream NLP benchmarking tasks were solveable by (more or less) overparameterised bag-of-words models, thereby adding to the literature calling into question their usefulness as benchmarks.

Chapter 5

In conclusion

I. Scientific outlook

Having gone into sufficient detail for each of the research threads we presented in this thesis, we are now well-placed to revisit our original research questions, and provide some form of answer to each of them. Summarising, therefore:

Research questions

1. **What can we say about the influences that differences in training or fine-tuning corpora have on the quality of multilingual language model spaces?**

   First, in Paper I, we show that even a few annotated fine-tuning instances can lead to substantial improvements on multilingual LM performance, on a range of downstream tasks. We also stress that the costs of annotation at these scales are low, and therefore recommend annotating wherever possible, even when doing so for large datasets is unrealistic.

   From Paper II, we can say that multilingual performance is affected by the degree of compression applied to corpus size distributions, even when controlling for the total number of tokens: a *moderate* level of compression, in our experiments, showed the best results downstream. A caveat is that these results may differ if one were to use different language models, or a different training setup.

   Finally, we show in Paper III that while cross-linguistic differences in multilingual space quality do appear to exist, a more significant contributing factor is differences at a *corpus* level: even when lemmatising (to ablate away morphology) and scrambling sentences (to ablate away syntax), models are capable of learning some degree of multilingualism; this ability shows correlations with, for instance, type-token ratios, or (negatively) to sentence length.

2. **What linguistic properties can we extract from multilingual language models?**

   In Paper IV, we adapt previously established tasks measuring linguistic competence to a multilingual setup. We probe English language encoders adapted to other languages via transfer learning; amongst other observations, we show that for certain phenomena, cross-linguistically transferred encoders can be *more* transparent than baseline English encoders.

In Paper V, we extend this setup to analyse deep pretrained language models we then move on to probing existing multilingual language models. We detail our findings and analyse them along the lenses of language, encoder and task; one noteworthy observation is the absence of any gap in performance between English and our other languages.

In Paper VI, we address work discussing the extractability of syntactic structure from language models, and flip the question to address how the choice of syntactic formalism affects this extractability across languages. We show that there are clearly visible differences, such as that shallower trees are easier to extract, and that the model thus prefers formalisms and languages that feature shallower trees.

3. **How do specific components within transformer-based language models act in multilingual contexts?**

   First, in Paper VII, we analyse precisely how attention weights drift when we fine-tune on annotated dependency syntax. We show that, uniformly across languages, syntactic trees become easier to extract from attention weights. We also show that, counterintuitively, if one were to freeze certain components during fine-tuning, freezing the *value* components of the attention mechanism has a greater effect on the extractability of syntactic structure than freezing the key and query components that compose the attention weights.

   Next, in Paper VIII, we show that choosing simpler position embedding methods tends to result in better multilingual performance: over-engineered position embedding approaches drift further away from the idealised sinusoidal embeddings, and we hypothesise that these hurt the ability of language models to compose sentences, which consequently hurts their cross-lingual transfer capabilities.

   Finally, we conclude with Paper IX, where we address the active conversation around training language models on scrambled corpora; we show that even in these scenarios, there are several clues (such as BPE tokenisation) that allow language models to effectively utilise their position embeddings to learn to model word order. We further demonstrate that there exist downstream tasks for which word order is critical.

Research directions

Next, we zoom out and analyse the future research potential that we believe each of our research threads has. A common observation made by people working in (or tangential to) NLP is how fast the field is moving. In addition, a thread that has unified university-based NLP research groups is that, partially due to a lack of sufficient resources to carry out NLP research at scale, this rapid change in the field of NLP tends to be driven by the private sector – a gap that appears to be growing wider every year, with no signs of slowing down. This has only naturally led to many within the academic community questioning what exactly we can achieve with limited resources, and precisely what even the state of NLP research should look like, given that so many models in deployment have parameter counts that make their analysis prohibitively expensive[1].

We therefore describe precisely those avenues of research where we believe academia can provide insight that private research labs could (or would) not.

Multilingual spaces Suprisingly, given the popularity of multilingual language models, research into the emergence and nature of multilinguality has been relatively sparse – though it is plausible that this is due to the resource constraints with training independent full-scale multilingual language models. Our work, as well as much of the literature on multilingual spaces, is far from conclusive; there is still a lot we do not know about the nature of these spaces. A potential angle that we did not have the opportunity to evaluate is the use of multilingual spaces bootstrapped on *synthetic languages*; this ought to tie into the already-existing body of work analysing language modelling on synthetic languages (Ravfogel et al., 2019; White and Cotterell, 2021). Yet another interesting research domain in this vein is the study of the effect that multilingual space quality can have on *monolingual* tasks; although, particularly at academic budgets, it is worth being wary of *emergence* (Wei et al., 2022), a phenomenon wherein large language models begin exhibiting vastly improved capabilities relevant to some task at some unpredictable point in training.

Probing Much ado has been made about the inherent difficulty of probing; of the numerous challenges surrounding probe selection, probing task design, and even interpreting the results of probing tasks (see Belinkov (2022)). The increasing dominance of generative language models has also led to the emergence of a new frame of analysis: that of *prompting*. Rather than using parameterised probes to extract model information, these models are fed input strings and instructed to generate a certain output (Li et al., 2022); Liu et al. (2023) provide a survey of prompting methods. We believe that prompting is a worthy 'successor' to the probing paradigm, albeit still in its infancy; there is considerable potential for research work on consistency in prompting, prompting from multilingual perspectives, etc.

Another research domain we wish to emphasise is those specifically inspired by the cognitive sciences. Examples of early such works include Ettinger (2020); while these are still linguistic tasks, they are

[1] Note that these issues were very relevant even in the context of this thesis: consider, for instance, how our studies on multilinguality tended to focus on our training models with substantially fewer parameters than widely-used pretrained models (Ravishankar and Nivre, 2022; Ravishankar and Søgaard, 2021; Ravishankar et al., 2021b), or focused entirely on fine-tuning, a less compute-intensive operation (Lauscher et al., 2020; Ravishankar et al., 2021a)

drawn from existing cognitive datasets, and as such also indicate human performance on such tasks. Other works involve probing (or merely evaluating) language models that have cognitive biases injected into their learning or fine-tuning process (Abdou et al., 2021; Barrett et al., 2018; Schwartz et al., 2019); Hollenstein et al. (2019) consolidate many such works and analyse the effects of injective 'human' information, on a variety of downstream tasks.

Architectural analyses It is hard to deny that language modelling architectures over the last few years have relied less upon human or cognitive analogy, and more upon empiricism. Indeed, transformer-based language models have come a long way since Vaswani et al., and many innovations over the original model have been a result of experimentation, enabled also by the growth of the field of NLP in general. This somewhat haphazard mix of components appears to 'just work'; however, we know less and less about why it works, *how* it works, and whether indeed this is the best way to go about language modelling (Dong et al., 2021); while language modelling appears to show consistent improvements with scale, it is unclear how much of this is simply driven by larger training corpora. We posit that only principled analyses of the behaviour of language model components could help shed some light on their strengths and weaknesses, and help adequately shape the direction of future research.

II. Societal Outlook

Having discussed the technical implications of our work, we now focus on the social implications. Relevant to this is the role that big tech plays in setting the direction of NLP research today; the economic background to this lies in the years following the subprime mortgage crisis of 2008. The growth of big tech has partially been driven by consequent monetary policies like quantitative easing, which resulted in a decade and a half of near-zero interest rates. The resultant ease of borrowing has led to an almost unprecedented accumulation of capital in the tech sector in general (Fernandez et al., 2020) – capital accumulated through *financial* mechanisms, rather than through meaningful innovation. The resultant economic dominance of big tech in the past decade has therefore resulted in their driving fundamental AI/NLP research, through their economic capacity to pay for the infrastructure modern NLP needs, as well as to disburse grants and fellowships at scale. This backdrop is crucial to understand the potential harms that the deployment of multilingual language technology entails.

The growth of AI research at tech giants has been formidable; a subset of this growth, relevant to NLP, has been through enabling language technology for languages that have hitherto been neglected in the broader community (Joshi et al., 2020). Examples of this push include Alphabet's 1000 Languages[2] and

[2] https://blog.google/technology/ai/ways-ai-is-scaling-helpful/

Meta's No Language Left Behind[3] initiatives. These seems like a commendable goal, at least on the face of it: however, we cannot ignore the implications of this progress being driven by big tech; in that light, we describe the issues with the state of multilingual NLP, as it stands.

Consider, for instance, the existence of monopolistic platforms (like Facebook). The economic conditions of the past decade have enabled digital service providers to monopolise their sectors in the digital sphere – often through using accumulated capital to fund mergers and acquisitions of potential competitors, such as Meta's acquisition of WhatsApp and Instagram, or Alphabet's acquisition of YouTube. In addition, their monopolistic status as platforms allows them monopolistic access to user data; this allows for their growth through network effects, or simply through their ability to provide better services on account of the quantities of data they possess, leading to a feedback loop, as this allows them access to even more data: a phenomenon that has been described as *platform capitalism* (Srnicek, 2017). The fact that multilingual NLP research is substantially driven by Alphabet and Meta must, therefore, be examined against this backdrop. Recall that for both these corporations, the vast majority of their revenue comes from advertising: that is, from the sale of data extracted from users. In the context of multilingual NLP and their inclusivity goals, this scenario effectively means that language communities are encouraged to provide more of their own data to these platforms, acquiescing – often without informed consent (Andreotta et al., 2022) – to turning themselves into data points, to be sold to corporations[4].

Harvey (2017) expands upon the Marxian concept of primitive accumulation to describe *accumulation via dispossession*, wherein he posits that the concentration of capital in the hands of a few is enabled through the large-scale commodification and privatisation of originally publicly held assets. This lens has increasingly been applied to the commodification of *data* (Thatcher et al., 2016), to argue that platforms are currently monetised by commodifying the online activity of their users. The output of this activity is sold to advertisers, thereby producing surplus value (Ekman, 2012, §3.2); this surplus value is a byproduct of the appropriation of surplus *labour* (Marx, 1867/2004). Succinctly: we spend our time on platforms, and in doing so, generate wealth for other people. Within this mode of production, therefore, the push for broader language support can thus also be seen as inevitable, in light of the fact that most as-yet-unexploited sources of data come from speakers of underresourced languages, in markets that large platforms seek to expand into. This commodification of our social interactions is a reflection of the need for capital to constantly expand into domains and regions as-yet unassimilated into capitalism (Luxemburg, 1913/2015).

It is reasonable to make the case that the growth of multilingual NLP can be seen as a form of *dual-use* of technology, with both positive and negative

[3] https://ai.facebook.com/research/no-language-left-behind/

[4] Often, given sufficient penetration of platform technology into a society, even explicit non-consent becomes ineffective for individual anonymisation, as individual patterns can be inferred (Tufekci, 2019).

potential uses. It should certainly be uncontroversial to say that many deployed use-cases of NLP are legitimately useful, and that the expansion of NLP to cover as-yet underresourced languages certainly has the *potential* to benefit speakers of those languages. It is, however, becoming increasingly and uncomfortably foregrounded that under the current paradigm of technological growth, the negatives of the spread of multilingual NLP may outweigh the positives. For instance, Facebook as a platform sees dynamic growth in as-yet unexplored markets, often through the spread of platforms that Meta acquires (such as Instagram) and through the development of better language technology; simultaneously, having infiltrated these markets, Meta's "state-of-the art" multilingual models somehow manage to allow genocidal content to slip through the cracks in countries as diverse as Myanmar (Milmo, 2021) (where they have been sued for enabling the Rohingya genocide), Kenya (Miriri, 2022), India (Perrigo, 2020), and even Norway (Støyva, 2022) – a highly developed nation with robust institutions. Given what we know of the capabilities of multilingual language technology and Meta's focus on multilingual NLP, we cannot simply look at this as a consequence of the technology not "being there" yet: these hate speech oversights are simply *part of Meta's business model* (Lauer, 2021), as they are what drives engagement.

Future work on multilingual NLP must, therefore, be grounded in critique of precisely this tendency on part of big tech: while it is tempting to look at the growth of multilingual NLP as benevolent inclusivity, the fact that this inclusivity is driven by Alphabet and Facebook means that this technology potentially causes more problems than it solves.

III. The state of NLP discourse

At this point, we step away from the core analyses deriving from this thesis and draw the reader's attention to the state of NLP research as of early 2023. Observations about the speed of innovation in NLP have been validated yet again at precisely the time of our writing this thesis, with the release of closed-source massively-parameterised conversational models. In this section, we describe what we believe the future of the academic NLP research landscape *should* look like; in doing so, it is impossible for us to ignore the elephant in the room, i.e., models like OpenAI's ChatGPT.

At the time of our writing this work, conversational AI models like ChatGPT have suddenly been brought into the public consciousness, and despite it being somewhat tangential to our focus, the existence of these models will unquestionably significantly affect the NLP research horizon. The ability of these models to generate semi-factual information with near-human fluency has been received both with excitement and with alarm, for the potential societal harms this easy-to-generate content could entail. Already, much has been written precisely about these tradeoffs, with suggestions such as better governmental regulation for where these models can be used; we refer the reader to van Dis et al. (2023) for one such recent summary. In this section, we instead focus on what we believe to be the societal factors that lead to the concentrated investment and hype surrounding these models, and on the effect this has on academia.

In order to do so, we draw the reader's attention to what ChatGPT represents, not in terms of its text-generation capabilities, but in terms of the investment priorities of its funding parties. The enthusiasm surrounding research into these systems has two effects. First, large corporations and angel investors, having accumulated billions of dollars (often from their sale of user data), proceed to re-invest these profits into seeking to automate away precisely those forms of labour that are not assimilated into the system of *wage labour*. This, in turns, drives the increasing marginalisation of broader chunks of society into a precariat. Consider, for instance, the enthusiasm around ChatGPT as a 'replacement' for writers, or poets, or musicians (in parallel with image generation models replacing artists). The critical point here is not whether or not ChatGPT is *capable* of replacing these professions yet – it is that billions of dollars are being enthusiastically spent on technology that, under our current economic paradigm, serves to marginalise a growing body of workers and artists *en masse* into an ever-growing proletariat. Second, the incidental increases in labour productivity that this automation heralds for wage labourers are fundamentally undemocratic: they do not result in a genuine improvement in labour conditions, nor are the profits that automation entails socialised. To the contrary, they are co-opted by the capitalist class, while labour conditions consistently worsen, leading to the rapid acceleration of wealth inequality.

Next, we draw the reader's attention to how the material advantage that the haphazard deployment of AI presents to the capitalist class has a substantial effect on *culture*. The cultural hegemony of techno-optimistic investors serves to a) disseminate their market logic into the public consciousness, and b) make their values seem like 'neutral' scientific progress. In NLP/AI, this hegemony is maintained by a class of intellectuals vocal in AI culture, from movements as seemingly diverse as AI risk, or longtermist think-tanks. The increasing integration of this hegemonic class into the class of media intellectuals leads to further amplification and normalisation of their philosophy (Herman and Chomsky, 2010), and eventually, hype surrounding the use and deployment of these models begins to dominate even academic discourse. It is critical to be aware of *why* this happens; while debates regarding the ethics of NLP systems exist, we also ought to be more conscious of the sparsity of discourse regarding the influence of the *base* on the *superstructure*[5] that drives NLP research through its chokehold on AI culture. An inordinate focus on the culture wars that dominate discourse around large language models is ultimately counterproductive – in the end, this culture is a purely cynical attempt to further entrench the *material* relations enabled by the haphazard deployment of AI.

[5] The terms *base* and *superstructure* refer to the Marxian division of society (Marx, 1859 / 1978), the former relating to the mode of production in a society, and the latter describing societal institutions – culture, law, religion, etc. – that worked to justify the existence of the base; of a certain mode of production. Thus, the base would shape the superstructure, while the superstructure would maintain the base, through leveraging its hegemony (Gramsci, 1948 / 2007) over culture.

IV. The way forward

So where does this leave us? While the last two sections may have painted a pessimistic picture, our current trajectory is far from inevitable. On a scientific front, we see countless opportunities – many related to the research threads in this thesis – in fields ranging from explainability to cognitive science. We do not advocate for neo-Luddism; multilingual NLP has resulted in numerous very useful tools for speakers of underresourced languages, and intepretability studies on models have contributed substantially to our understanding of how these models function, which is a scientifically critical goal. At the same time, quiet acceptance of the inevitability of the large-scale deployment of these language models without regard to the consequences, economic or social, mirrors the broader societal internalisation of the inevitability of our present economic reality (Fisher, 2009), and is ultimately unhelpful.

Even though the resource gap between academia and the industry seems unlikely to close in the foreseeable future, we posit that there is substantial research potential within academia and describe some such avenues of research. We advocate for an academic shift away from the current paradigm of treating academia as state-funded training grounds for big tech corporations – incidentally, the very same corporations that prolifically avoid paying taxes (Regan, 2020). Further, we go as far as suggesting that there is no real point in academia trying to compete with the private sector: the proliferation of privately funded research labs is a relatively modern phenomenon, and particularly in research domains where fundamental research can potentially result in marketable products, it is abundantly clear that academia cannot hope to close the gap with the private sector – at the very least, not without becoming a tax-funded industry outpost.

Instead, we advocate for a clearer demarcation of the roles of academia and the industry in the context of NLP, and advocate for increased academic engagement with civil society, and increased communication with policymakers, legal experts, journalists, etc. Very often, there is a disconnect between governmental AI regulatory authorities, or even independent AI watchdogs, and actual AI research, particularly where model intepretability is concerned: this disconnect has resulted in relatively ignorant policy positions like advocacy for "algorithmic transparency", which is meaningless in the context of large models that infer based on the biases present in their gigabytes of uncurated training data. Further, while there is already much academic discussion surrounding the dangers and potential harms of the deployment of ChatGPT-scale language models, it appears to be mostly restricted to academic silos, with little cross-pollination with legal and policy experts. We therefore advocate for the future of NLP academia to fill precisely this niche, and to improve communication between the research community and regulatory spaces, in order for civil society to better understand the impact of the deployment of AI on itself.

Bibliography

Abdou, M. et al. (2019). “Higher-order Comparisons of Sentence Encoder Representations”. 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)*.

Abdou, M. et al. (2021). “Does Injecting Linguistic Structure into Language Models Lead to Better Alignment with Brain Recordings?” In: no. arXiv:2101.12608.

Abdou, M. et al. (2022). “Word order does matter and shuffled language models know it”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Abnar, S. and Zuidema, W. (2020). “Quantifying Attention Flow in Transformers”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Adi, Y. et al. (2017). “Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Alleman, M. et al. (2021). “Syntactic Perturbations Reveal Representational Correlates of Hierarchical Phrase Structure in Pretrained Language Models”. In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics.

Andreotta, A. J. et al. (2022). “AI, Big Data, and the Future of Consent”. In: *AI & SOCIETY* vol. 37, no. 4.

Artetxe, M. and Schwenk, H. (2019). “Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond”. In: vol. 7. Cambridge, MA: MIT Press.

Artetxe, M. et al. (2018a). “A Robust Self-Learning Method for Fully Unsupervised Cross-Lingual Mappings of Word Embeddings”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

— (2018b). “Generalizing and Improving Bilingual Word Embedding Mappings with a Multi-Step Framework of Linear Transformations”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 32, no. 1.

Artetxe, M. et al. (2018c). “Unsupervised Neural Machine Translation”. In: *International Conference on Learning Representations*.

Artetxe, M. et al. (2020). “On the cross-lingual transferability of monolingual representations”. In: *Proceedings of ACL*.

Ba, J. L. et al. (2016). “Layer Normalization”. In: *stat* vol. 1050.

Bahdanau, D. et al. (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: ed. by Bengio, Y. and LeCun, Y.

Barrett, M. et al. (2018). "Sequence Classification with Human Attention". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics.

Bastings, J. and Filippova, K. (2020). "The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?" In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics.

Belinkov, Y. (2022). "Probing Classifiers: Promises, Shortcomings, and Advances". In: *Computational Linguistics* vol. 48, no. 1.

Belinkov, Y. et al. (2017). "What do Neural Machine Translation Models Learn about Morphology?" en. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics.

Bengio, Y. et al. (2000). "A Neural Probabilistic Language Model". In: *Advances in Neural Information Processing Systems*. Ed. by Leen, T. et al. Vol. 13. MIT Press.

Bojanowski, P. et al. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* vol. 5.

Bowman, S. R. and Dahl, G. (2021). "What Will it Take to Fix Benchmarking in Natural Language Understanding?" 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.

Brown, T. et al. (2020). *Language Models are Few-Shot Learners*. Ed. by Larochelle, H. et al.

Brunner, G. et al. (2020). "On Identifiability in Transformers". In: *International Conference on Learning Representations*.

Cao, S. et al. (2020). "Multilingual alignment of contextual word representations". In: *Proceedings of ICLR*.

Chi, E. A. et al. (2020). "Finding Universal Grammatical Relations in Multilingual BERT". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Chrupała, G. and Alishahi, A. (2019). "Correlating Neural and Symbolic Representations of Language". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Chu, Y.-J. (1965). "On the shortest arborescence of a directed graph". In: *Scientia Sinica* vol. 14.

Church, K. W. (2017). "Emerging Trends: I Did It, I Did It, I Did It, But. . ." In: *Natural Language Engineering* vol. 23, no. 3.

Chaabouni, R. et al. (2020). "Compositionality and Generalization In Emergent Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Clark, K. et al. (2019). "What Does BERT Look at? An Analysis of BERT's Attention". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics.

Clouatre, L. et al. (2022). "Local Structure Matters Most: Perturbation Study in NLU". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics.

Conneau, A. et al. (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics.

Conneau, A. et al. (2018a). "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Conneau, A. et al. (2018b). "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.

Conneau, A. et al. (2020). "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.

Devlin, J. et al. (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.

Dong, Y. et al. (2021). "Attention is not all you need: Pure attention loses rank doubly exponentially with depth". In: *International Conference on Machine Learning*. PMLR.

Dufter, P. and Schütze, H. (2020). "Identifying Elements Essential for BERT's Multilinguality". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Dufter, P. et al. (2022). "Position Information in Transformers: An Overview". In: vol. 48. 3. Cambridge, MA: MIT Press.

Edmiston, D. (2020). "A Systematic Analysis of Morphological Content in BERT Models for Multiple Languages". In: no. arXiv:2004.03032.

Ekman, M. (2012). "Understanding accumulation: The relevance of Marx's theory of primitive accumulation in media and communication studies". In: *tripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society* vol. 10, no. 2.

Ettinger, A. (2020). "What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models". In: *Transactions of the Association for Computational Linguistics* vol. 8.

Fernandez, R. et al. (2020). “The financialisation of Big Tech”. In: *SOMO (Stichting Onderzoek Multinationale Ondernemingen)*.

Firth, R. J. (1957). “A Synopsis of Linguistic Theory, 1930-1955”. In: *Studies in Linguistic Analysis*.

Fisher, M. (2009). *Capitalist realism: Is there no alternative?* John Hunt Publishing.

Gramsci, A. (1948 / 2007). “Selections from the Prison Notebooks”. In: *On Violence*. Duke University Press.

Gupta, A. et al. (2015). “Distributional Vectors Encode Referential Attributes”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics.

Harvey, D. (2017). “The’new’imperialism: accumulation by dispossession”. In: *Karl Marx*. Routledge.

Haviv, A. et al. (2022). “Transformer Language Models without Positional Encodings Still Learn Positional Information”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

He, P. et al. (2021). “DeBERTa: Decoding-Enhanced BERT with Disentangled Attention”. In: *International Conference on Learning Representations*.

Herman, E. S. and Chomsky, N. (2010). *Manufacturing consent: The political economy of the mass media*. Random House.

Hewitt, J. and Liang, P. (2019). “Designing and Interpreting Probes with Control 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.

Hewitt, J. and Manning, C. D. (2019). “A structural probe for finding syntax in word representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Hofmann, V. et al. (2020). *DagoBERT: Generating Derivational Morphology with a Pretrained Language Model*. Online.

Hollenstein, N. et al. (2019). *Advancing NLP with Cognitive Language Processing Signals*.

Hu, J. et al. (2020). “XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation”. In: Proceedings of Machine Learning Research vol. 119. Ed. by III, H. D. and Singh, A.

Huang, Z. et al. (2020). “Improve Transformer Models with Better Relative Position Embeddings”. In.

Hupkes, D. et al. (2018). “Visualisation and ’diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure”. In: *Journal of Artificial Intelligence Research* vol. 61.

Hupkes, D. et al. (2020). “Compositionality Decomposed: How do Neural Networks Generalise? (Extended Abstract)”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Bessiere, C. Journal track. International Joint Conferences on Artificial Intelligence Organization.

Jacovi, A. and Goldberg, Y. (2020). "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Jain, S. and Wallace, B. C. (2019). "Attention is not Explanation". 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)*.

Jawahar, G. et al. (2019). "What Does BERT Learn about the Structure of Language?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Jia, R. and Liang, P. (2017). *Adversarial Examples for Evaluating Reading Comprehension Systems*. Copenhagen, Denmark.

Joshi, P. et al. (2020). "The State and Fate of Linguistic Diversity and Inclusion in the NLP World". In: *Proceedings of ACL*.

K, K. et al. (2020). "Cross-Lingual Ability of Multilingual BERT: An Empirical Study". In: *International Conference on Learning Representations*.

Ke, G. et al. (2020). "Rethinking Positional Encoding in Language Pre-Training". In: *International Conference on Learning Representations*.

Kiros, R. et al. (2015). "Skip-Thought Vectors". In: *Advances in Neural Information Processing Systems* vol. 28.

Kobayashi, G. et al. (2020). In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Kornblith, S. et al. (2019). "Similarity of neural network representations revisited". In.

Kriegeskorte, N. et al. (2008). "Representational similarity analysis-connecting the branches of systems neuroscience". In: *Frontiers in systems neuroscience* vol. 2.

Krishna, S. et al. (2022). "The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective". In: vol. abs/2202.01602. arXiv: 2202.01602.

Kulmizev, A. et al. (2020a). "Do Neural Language Models Show Preferences for Syntactic Formalisms?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

— (2020b). "Do Neural Language Models Show Preferences for Syntactic Formalisms?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Lample, G. and Conneau, A. (2019). "Cross-Lingual Language Model Pretraining". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lample, G. et al. (2018a). "Unsupervised Machine Translation Using Monolingual Corpora Only". In: *International Conference on Learning Representations*.

Lample, G. et al. (2018b). "Word Translation Without Parallel Data". In: *International Conference on Learning Representations*.

Lauer, D. (2021). “Facebook’s ethical failures are not accidental; they are part of the business model”. In: *AI and Ethics* vol. 1, no. 4.

Lauscher, A. et al. (2020). “From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Li, J. et al. (2022). *Probing via Prompting*.

Liu, N. F. et al. (2019a). “Linguistic Knowledge and Transferability of Contextual Representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Liu, P. et al. (2023). “Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ACM Computing Surveys* vol. 55, no. 9.

Liu, Y. et al. (2019b). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv:1907.11692*.

Luong, T. et al. (2015). “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics.

Luxemburg, R. (1913/2015). *The accumulation of capital*. Routledge.

Marx, K. (1867/2004). *Capital: Volume I*. Vol. 1. Penguin UK.

— (1859 / 1978). “Preface to a Contribution to the Critique of Political Economy”. In: *The Marx-Engels Reader* vol. 2.

McCoy, T. et al. (2019). “Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference”. In.

Mikolov, T. et al. (2013a). “Efficient Estimation of Word Representations in Vector Space”. In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Bengio, Y. and LeCun, Y.

Mikolov, T. et al. (2013b). “Exploiting similarities among languages for machine translation”. In: *CoRR* vol. abs/1309.4168. arXiv: 1309.4168.

Milmo, D. (2021). “Rohingya Sue Facebook for £150bn over Myanmar Genocide”. In: *The Guardian*.

Miriri, D. (2022). “Kenya Orders Meta’s Facebook to Tackle Hate Speech or Face Suspension”. In: *Reuters*.

Mollica, F. et al. (2020). “Composition is the core driver of the language-selective network”. In: *Neurobiology of Language* vol. 1, no. 1.

Nivre, J. et al. (2017). *Universal Dependencies 2.1*.

Pascual, D. et al. (2021). “Telling BERT’s Full Story: From Local Attention to Global Aggregation”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics.

Pennington, J. et al. (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.

Perrigo, B. (2020). “Facebook’s Ties to India’s Ruling Party Complicate Its Fight Against Hate Speech”. In: *Time*.

Peters, M. et al. (2018a). “Dissecting Contextual Word Embeddings: Architecture and Representation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Peters, M. E. et al. (2018b). “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.

Pimentel, T. et al. (2020a). “Information-Theoretic Probing for Linguistic Structure”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Pimentel, T. et al. (2020b). “Pareto Probing: Trading Off Accuracy for Complexity”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Pires, T. et al. (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.

Poliak, A. et al. (2018). “Hypothesis only baselines in natural language inference”. In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics.

Press, O. et al. (2021). “Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation”. In: *International Conference on Learning Representations*.

Radford, A. et al. (2018). *Improving language understanding by generative pre-training*.

Radford, A. et al. (2019). “Language models are unsupervised multitask learners”. In: *OpenAI Blog* vol. 1, no. 8.

Raffel, C. et al. (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: vol. 21. 1. JMLRORG.

Raganato, A., Tiedemann, J., et al. (2018). “An analysis of encoder representations in transformer-based machine translation”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics.

Raghu, M. et al. (2017). “SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability”. In: *Advances in Neural Information Processing Systems*.

Rahimi, A. et al. (2019). “Massively Multilingual Transfer for NER”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Ravfogel, S. et al. (2019). “Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages”. In: *Proceedings of NAACL-HLT*.

Ravishankar, V. and Nivre, J. (2022). “The Effects of Corpus Choice and Morphosyntax on Multilingual Space Induction”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Ravishankar, V. and Søgaard, A. (2021). “The Impact of Positional Encodings on Multilingual Compression”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Ravishankar, V. et al. (2019a). “Multilingual Probing of Deep Pre-Trained Contextual Encoders”. In: *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*. Turku, Finland: Linköping University Electronic Press.

Ravishankar, V. et al. (2019b). “Multilingual probing of deep pre-trained contextual encoders”. In: *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*.

Ravishankar, V. et al. (2019c). “Probing Multilingual Sentence Representations With X-Probe”. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics.

Ravishankar, V. et al. (2021a). “Attention Can Reflect Syntactic Structure (If You Let It)”. In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics.

Ravishankar, V. et al. (2021b). “Multilingual ELMo and the Effects of Corpus Sampling”. In: *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden.

Regan, A. (2020). “Until the EU Tackles Tax Avoidance, Big Companies Will Keep Getting Away with It”. In: *The Guardian*.

Ribeiro, M. T. et al. (2020). “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Rogers, A. et al. (2020). “A Primer in BERTology: What we know about how BERT works”. In: vol. 8. Cambridge, MA: MIT Press.

Ruder, S. et al. (2019). “A survey of cross-lingual embedding models”. In: *Journal of Artificial Intelligence Research* vol. 65.

Rush, A. M. et al. (2015). *A Neural Attention Model for Abstractive Sentence Summarization*. Lisbon, Portugal.

Saphra, N. and Lopez, A. (2019). “Understanding Learning Dynamics Of Language Models with SVCCA”. 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.

Schwartz, D. et al. (2019). “Inducing brain-relevant bias in natural language processing models”. In: *Advances in neural information processing systems* vol. 32.

Sennrich, R. et al. (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.

Seo, M. et al. (2017). “Bidirectional Attention Flow for Machine Comprehension”. In: *International Conference on Learning Representations*.

Serrano, S. and Smith, N. A. (2019). “Is Attention Interpretable?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Shaw, P. et al. (2018). “Self-Attention with Relative Position Representations”. In.

Sinha, K. et al. (2021a). “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Sinha, K. et al. (2021b). “Unnatural Language Inference”. 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.

Srnicek, N. (2017). *Platform capitalism*. John Wiley & Sons.

Støyva, A. (2022). “De testet Facebooks håndtering av hatefullt innhold. Amnesty og Forbrukerrådet mener resultatet er sjokkerende.” In: *Aftenposten*.

Su, J. et al. (2022). “RoFormer: Enhanced Transformer with Rotary Position Embedding”. In: no. arXiv:2104.09864.

Sun, K. and Marasović, A. (2021). “Effective Attention Sheds Light On Interpretability”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics.

Søgaard, A. et al. (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.

Tenney, I. et al. (2019a). “BERT Rediscovers the Classical NLP Pipeline”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Tenney, I. et al. (2019b). “What do you learn from context? Probing for sentence structure in contextualized word representations”. In: *International Conference on Learning Representations*.

Thatcher, J. et al. (2016). “Data colonialism through accumulation by dispossession: New metaphors for daily data”. In: *Environment and Planning D: Society and Space* vol. 34, no. 6.

Tufekci, Z. (2019). “Opinion | Think You’re Discreet Online? Think Again”. In: *The New York Times*.

van Dis, E. A. M. et al. (2023). “ChatGPT: Five Priorities for Research”. In: *Nature* vol. 614, no. 7947.

Vaswani, A. et al. (2017a). “Attention Is All You Need”. In: *arXiv:1706.03762 [cs]*.

Vaswani, A. et al. (2017b). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al. Vol. 30. Curran Associates, Inc.

Vig, J. and Belinkov, Y. (2019). “Analyzing the structure of attention in a transformer language model”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics.

Voita, E. and Titov, I. (2020). “Information-Theoretic Probing with Minimum Description Length”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Vulić, I. et al. (2019). “Do We Really Need Fully Unsupervised Cross-Lingual Embeddings?” 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.

Vulić, I. et al. (2020a). “Are All Good Word Vector Spaces Isomorphic?” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Vulić, I. et al. (2020b). “Probing Pretrained Language Models for Lexical Semantics”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Wang, A. et al. (2018). “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics.

Wang, B. and Komatsuzaki, A. (2021). *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. https://github.com/kingoflolz/mesh-transformer-jax.

Wang, B. et al. (2021). “On Position Embeddings in BERT”. In: *International Conference on Learning Representations*.

Wang, X. et al. (2019). “Self-Attention with Structural Position Representations”. 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.

Wei, J. et al. (2022). “Emergent Abilities of Large Language Models”. In: *Transactions on Machine Learning Research*.

White, J. C. and Cotterell, R. (2021). “Examining the Inductive Bias of Neural Language Models with Artificial Languages”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics.

Wiedemann, G. et al. (2019). “Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings”. In: no. arXiv:1909.10430.

Wiegreffe, S. and Pinter, Y. (2019). “Attention is not not Explanation”. 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)*.

Wu, S. and Dredze, M. (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.

Wu, S. et al. (2020a). “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.

Wu, Y. et al. (2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *CoRR*, no. arXiv:1609.08144.

Wu, Z. et al. (2020b). “Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Yang, Y. et al. (2019). “PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics.

Yang, Z. et al. (2016). “Hierarchical Attention Networks for Document Classification”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics.

Zhang, K. W. and Bowman, S. R. (2019). *Language Modeling Teaches You More Syntax than Translation Does: Lessons Learned Through Auxiliary Task Analysis*.

Zhang, M. et al. (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.

Zhu, Z. and Rudzicz, F. (2020). “An Information Theoretic View on Selecting Linguistic Probes”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Papers

Paper I

From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers

*Anne Lauscher, *Vinit Ravishankar, Goran Glavaš, Ivan Vulić

Appears in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, November 2020, pp. 4483–4499. DOI: 10.18653/v1/2020.emnlp-main.363.

Abstract

Massively multilingual transformers (MMTs) pretrained via language modeling (e.g., mBERT, XLM-R) have become a default paradigm for zero-shot language transfer in NLP, offering unmatched transfer performance. Current evaluations, however, verify their efficacy in transfers (a) to languages with sufficiently large pretraining corpora, and (b) between close languages. In this work, we analyze the limitations of downstream language transfer with MMTs, showing that, much like cross-lingual word embeddings, they are substantially less effective in resource-lean scenarios and for distant languages. Our experiments, encompassing three lower-level tasks (POS tagging, dependency parsing, NER) and two high-level tasks (NLI, QA), empirically correlate transfer performance with linguistic proximity between source and target languages, but also with the size of target language corpora used in MMT pretraining. Most importantly, we demonstrate that the inexpensive few-shot transfer (i.e., additional fine-tuning on a few target-language instances) is surprisingly effective across the board, warranting more research efforts reaching beyond the limiting zero-shot conditions.

Contents

I.1 Introduction and Motivation

Labeled datasets of sufficient size support supervised learning in NLP. The notorious tediousness, subjectivity, and cost of linguistic annotation (Dandapat et al., 2009; Fort, 2016; Sabou et al., 2012), coupled with plethora of structurally different NLP tasks, lead to existence of such datasets only for a handful of resource-rich languages (Bender, 2011; Joshi et al., 2020; Ponti et al., 2019). This data scarcity renders the need for effective *cross-lingual transfer* strategies: how can we exploit abundant labeled data from resource-rich languages to make predictions in resource-lean languages? In the most extreme scenario, termed *zero-shot cross-lingual transfer*, not a single labeled instance exists for a target language. Recent work has placed much emphasis on this scenario exactly; in theory, it offers the widest portability across the world's 7,000+ languages (Artetxe et al., 2020b; Cao et al., 2020; Hu et al., 2020; Lin et al., 2019; Pires et al., 2019).

The current mainstay of cross-lingual transfer in NLP are approaches based on continuous cross-lingual representation spaces such as cross-lingual word embeddings (CLWEs) (Ruder et al., 2019) and, most recently, massively multilingual transformer networks (MMTs), pretrained on multilingual corpora with language modeling (LM) objectives (Conneau et al., 2020; Devlin et al., 2019; Lample and Conneau, 2019). The latter have *de facto* become the default language transfer paradigm, with multiple studies reporting their unparalleled transfer performance Karthikeyan et al., 2020; Pires et al., 2019; Rönnqvist et al., 2019; Wu and Dredze, 2019; Wu et al., 2020.

Key Questions and Contributions. In this work, we dissect the current state-of-the-art MMT-based approach to (zero-shot) cross-lingual transfer, and analyze a variety of conditions and factors that critically impact or limit effective cross-lingual transfer. Our aim is to provide answers to the following crucial questions.

(Q1) *What is the role of language (dis)similarity and language-specific corpora size in pretraining?*

Current cross-lingual transfer via MMTs is still primarily focused on either (1) languages that are typologically or etymologically close to English (e.g., German, Scandinavian languages, French, Spanish), or (2) languages with large monolingual corpora, well-represented in the multilingual pretraining corpora (e.g., Arabic, Hindi, Chinese). Wu et al. (2020) suggest that LM-pretrained transformers, much like static word embeddings models, produce topologically similar representation spaces that can easily be aligned between languages, offering this as explanation of language transfer efficacy of MMTs. However, transfer with static CLWEs has been shown ineffective between dissimilar languages (Søgaard et al., 2018; Vulić et al., 2019) or languages with small corpora (Vulić et al., 2020).

We thus scrutinize MMTs in diverse zero-shot transfer settings and find, in line with prior work on CLWEs, that MMTs' transfer performance critically depends on (1) linguistic (dis)similarity between the source and target language and (2) size of the pretraining corpus of the target language.

(Q2) *What is the role of a particular task in consideration for transfer performance?*

We conduct all analyses across five different tasks, which we roughly divide into two groups: (1) "low-level" tasks (POS-tagging, dependency parsing, and NER); and (2) "high-level" language understanding (LU) tasks (NLI and QA). We show that transfer performance in both zero-shot and few-shot scenarios largely depends on the "task level".

(Q3) *Can we (even) predict transfer performance?*

Running a simple regression on available transfer results, we show that we can (roughly) predict the transfer performance from (1) language proximity (Littell et al., 2017) for low-level tasks; (2) combination of language proximity and size of target-language pretraining corpora for high-level tasks.

(Q4) *Should we focus more on few-shot transfer scenarios and quick annotation cycles?*

Complementing the efforts on improving zero-shot transfer (Cao et al., 2020), we point to few-shot transfer as a very effective mechanism for improving target-language performance. Similar to the seminal "pre-neural" work of Garrette and Baldridge (2013), our results suggest that only several hours (or even minutes) of annotation work can "buy" substantial performance gains for low-resource targets. For all five tasks in our study, we obtain substantial (and in some cases surprisingly large) improvements with minimal annotation effort. For instance, we improve dependency parsing for some target languages up to 40 UAS points with as few as 10 target language sentences. Crucially, the few-shot gains are most pronounced exactly where zero-shot transfer fails: for distant target languages with small monolingual corpora.

I.2 Background and Related Work

For completeness, we provide a brief overview of **1)** cross-lingual transfer approaches, with a focus on **2)** massively multilingual transformer (MMT) models, and then **3)** position our work w.r.t. other studies that examine different properties of MMTs.

I.2.1 Cross-Lingual Transfer Paradigms

Language transfer entails representing texts from both the source and target language in a shared cross-lingual space. Transfer paradigms based on discrete text representations include *machine translation* (MT) of target language text to the source language (or vice-versa) (Eger et al., 2018; Mayhew et al., 2017), and grounding texts from both languages in *multilingual knowledge bases* (KBs)

(Lehmann et al., 2015; Navigli and Ponzetto, 2012). While reliable MT hinges on availability of large parallel corpora, transfer via multilingual KBs (Camacho-Collados et al., 2016; Mrkšić et al., 2017) is impaired by the limited KB coverage and inaccurate entity linking (Moro et al., 2014; Raiman and Raiman, 2018).

Therefore, recent years have seen a surge of language transfer methods based on continuous representation spaces. The previous state-of-the-art, cross-lingual word embeddings (CLWEs) (Ammar et al., 2016; Artetxe et al., 2017; Glavaš et al., 2019; Mikolov et al., 2013; Smith et al., 2017; Vulić et al., 2019) and sentence embeddings (Artetxe and Schwenk, 2019), have most recently been replaced by massively multilingual transformers (MMTs) pretrained with LM objectives (Conneau et al., 2020; Devlin et al., 2019; Lample and Conneau, 2019).

I.2.2 Massively Multilingual Transformers

Multilingual BERT (mBERT). At BERT's (Devlin et al., 2019) core is a multi-layer transformer network (Vaswani et al., 2017), parameters of which are pretrained using masked language modeling (MLM) and next sentence prediction (NSP). In MLM, some tokens are masked out and they need to be recovered from the context; NSP predicts adjacency of sentences in text, informing the transformer of longer dependencies, beyond sentence boundaries. Liu et al. (2019) introduce RoBERTa, a more robust instance of BERT trained on larger corpora using only the MLM objective. Multilingual BERT (mBERT) is an instance of BERT trained on concatenation of 104 largest Wikipedias. The effects of underfitting for languages with small Wikipedias and overfitting to languages with large Wikipedias, are respectively attenuated with exponentially smoothed up-sampling and down-sampling.

XLM on RoBERTa (XLM-R). XLM-R (Conneau et al., 2020) is an instance of RoBERTa, robustly trained on a large multilingual CommonCrawl-100 (CC-100) corpus (Wenzek et al., 2019) covering 100 languages. mBERT's corpus and CC-100 share 88 languages, with corresponding CC-100's portions being much larger than mBERT's Wikipedias.

The "Curse of Multilinguality". For XLM-R, Conneau et al. (2020) observe that for a fixed model capacity, downstream cross-lingual transfer improves with more pretraining languages up to a point after which adding more pretraining languages hurts downstream transfer. This effect, termed the "curse of multilinguality", can be mitigated by increasing model's capacity (Artetxe et al., 2020b) or additional training for particular language pairs (Pfeiffer et al., 2020). This points to MMTs' capacity (i.e., computational budgets), as a critical factor for effective zero-shot transfer.

In contrast, we identify few-shot transfer as a much more cost-effective strategy for improving downstream target language performance (§I.4). We show for a number of target languages and downstream tasks, that one can obtain large performance gains at very small annotation cost, without having to pretrain from scratch an MMT of larger capacity.

I.2.3 Cross-Lingual Transfer with MMTs

A body of recent work probed the knowledge encoded in MMTs, primarily mBERT. Libovickỳ et al. (2020) analyze language-specific versus language-universal knowledge encoded in mBERT. Pires et al. (2019) demonstrate mBERT to be effective for POS-tagging and NER zero-shot transfer between related languages. Wu and Dredze (2019) extend this analysis to more tasks and languages, and show that mBERT-based transfer is on a par with the best task-specific zero-shot transfer approaches. Similarly, Karthikeyan et al. (2020) prove mBERT to be effective for NER and NLI transfer to Hindi, Spanish, and Russian.[1] Importantly, they show that transfer effectiveness does not depend on the vocabulary overlap between the languages.

In most recent work, concurrent to this, Hu et al. (2020) introduce XTREME, a benchmark for evaluating multilingual encoders encompassing 9 tasks and 40 languages.[2] While the primary focus is a large-scale zero-shot transfer evaluation, they also experiment with target-language fine-tuning (1,000 instances for POS and NER). While Hu et al. (2020) focus on the evaluation aspects and protocols, in this work, we provide a more detailed analysis of the factors that hinder effective zero-shot transfer across several tasks.[3] We also put more emphasis on few-shot transfer, and approach it differently: by sequentially fine-tuning MMTs, first on (larger) source language training data and then on few target-language instances.

Artetxe et al. (2020b) and Wu et al. (2020) analyze different monolingual BERTs to explain transfer efficacy of mBERT. They find topological similarities between monolingual spaces, suggesting these are responsible for effective language transfer with MMTs. In essence, their work recasts the well-known assumption of approximate isomorphism of monolingual representation spaces (Søgaard et al., 2018). For CLWEs, this assumption does not hold for distant languages (Søgaard et al., 2018; Vulić et al., 2019), and in face of monolingual corpora of small size (Vulić et al., 2020). We demonstrate that the same is the case for zero-shot language transfer with MMTs: target-language performance drastically decreases as we move to more distant target languages with smaller pretraining corpora.

I.3 Zero-Shot Transfer: Analyses

We first address Q1 and Q2 (see §I.1): we conduct zero-shot language transfer experiments for five different tasks and analyze the factors behind the varying performance drops across target languages.

[1] Note that all three are high-resource Indo-European languages with large Wikipedias.

[2] Note that none of the individual tasks in XTREME covers all 40 languages, but much smaller language subsets.

[3] We leave an even more general analysis that combines transfer both across tasks (Glavaš and Vulić, 2020; Pruksachatkun et al., 2020) *and* across languages for future work.

I.3.1 Experimental Setup

Tasks and Languages. We experiment with – **a)** low-level structured prediction tasks: POS-tagging, dependency parsing, and NER and **b)** high-level language understanding (LU) tasks: NLI and QA. We investigate if the factors that drive transfer performance differ between the two task groups.

Dependency Parsing (DEP). We use Universal Dependency treebanks (UD, Nivre et al., 2017) for English and following target languages (from 8 language families): Arabic (AR), Basque (EU), (Mandarin) Chinese (ZH), Finnish (FI), Hebrew (HE), Hindi (HI), Italian (IT), Japanese (JA), Korean (KO), Russian (RU), Swedish (SV), and Turkish (TR).

Part-of-speech Tagging (POS). Again, we use UD and obtain the Universal POS-tag (UPOS) annotations from the same treebanks as with DEP.

Named Entity Recognition (NER). We resort to the NER WikiANN dataset from Rahimi et al. (2019). We experiment with the same set of 12 target languages as in DEP and POS.

Cross-lingual Natural Language Inference (XNLI). We evaluate on the XNLI corpus (Conneau et al., 2018) created by translating dev and test portions of the English Multi-NLI dataset (Williams et al., 2018) into 14 languages by professional translators (French (FR), Spanish (ES), German (DE), Greek (EL), Bulgarian (BG), Russian (RU), Turkish (TR), Arabic (AR), Vietnamese (VI), Thai (TH), Chinese (ZH), Hindi (HI), Swahili (SW), and Urdu (UR)).

Cross-lingual Question Answering (XQuAD). We rely on the XQuAD dataset (Artetxe et al., 2020b), created by translating the 240 dev paragraphs (from 48 documents) and corresponding 1,190 QA pairs of SQuAD v1.1 (Rajpurkar et al., 2016) to 11 languages (ES, DE, EL, RU, TR, AR, VI, TH, ZH, and HI). In order to allow for a comparison between zero-shot and few-shot transfer (see §I.4), we reserve 10 documents as the development set for our experiments and evaluate on the remaining 38 articles.[4]

Fine-tuning. For higher-level tasks, we perform standard downstream fine-tuning of LM-pretrained mBERT and XLM-R. For lower-level tasks, we instead freeze the transformer and train only task-specific classifiers.[5,6]

We add the following task-specific architectures on top of MMTs: for DEP we add the biaffine parsing head (Dozat and Manning, 2017; Kondratyuk and Straka, 2019); for POS, we attach a simple feed-forward token-level classifier; for

[4] As a general note, while the effects of "translationese" might have some impact on the absolute numbers (Artetxe et al., 2020c), they are not prominent enough to have any impact on the relative trends in the reported results (e.g., zero-shot vs. few-shot performance). For both XNLI and XQuAD, the translations were done completely manually and not via post-editing of MT (which would pose a higher "translationese" risk). Moreover, having an independently created test set in each language would impede comparability across languages.

[5] This gave slightly better performance than fine-tuning.

[6] We tokenize the input for each model with the corresponding pretrained fixed-vocabulary tokenizer: WordPiece tokenizer (Wu et al., 2016) with the vocabulary of 110K tokens for mBERT, and the SentencePiece BPE tokenizer (Sennrich et al., 2016) with the vocabulary of 250K tokens for XLM-R.

		EN	ZH	TR	RU	AR	HI	EU	FI	HE	IT	JA	KO	SV	VI	TH	ES	EL	DE	FR	BG	SW	UR
Task	**Model**		Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
DEP	B	91.2	-43.9	-46.0	-28.1	-56.4	-36.1	-50.2	-30.7	-36.1	-17.1	**-60.1**	-56.1	-14.3	-	-	-	-	-	-	-	-	-
	X	92.0	**-85.4**	-44.2	-29.7	-54.6	-39	-49.5	-26.7	-39	-23.5	-80.5	-56.0	-16.3	-	-	-	-	-	-	-	-	-
POS	B	95.8	-38.0	-35.9	-16.0	-40.1	-33.4	-34.6	-21.9	-33.4	-19.8	**-46.1**	-42.0	-9.6	-	-	-	-	-	-	-	-	-
	X	96.3	-69.2	-27.7	-14.3	-37.1	-27.3	-31.9	-17.9	-27.3	-19.0	**-77.0**	-37.3	-10.7	-	-	-	-	-	-	-	-	-
NER	B	92.4	-23.3	-11.6	-10.7	**-31.7**	-11.1	-12.8	-3.8	-11.1	-2.6	-25.7	-13.8	-6.7	-	-	-	-	-	-	-	-	-
	X	91.6	**-34.8**	-6.2	-13.7	-24.6	-16.5	-8.0	-0.9	-16.5	-2.4	-30.1	-15.6	-2.2	-	-	-	-	-	-	-	-	-
XNLI	B	82.8	-13.6	-20.6	-13.5	-17.3	-21.3	-	-	-	-	-	-	-	-11.9	-28.1	-8.1	-14.1	-10.5	-7.8	-13.3	**-33.0**	-23.4
	X	84.3	-11.0	-11.3	-9.0	-13.0	-14.2	-	-	-	-	-	-	-	-9.7	-12.3	-5.8	-8.9	-7.8	-6.1	-6.6	**-20.2**	-17.3
XQuAD	B	71.1	-22.9	-34.2	-19.2	-24.7	-28.6	-	-	-	-	-	-	-	-22.1	**-43.2**	-16.6	-28.2	-14.8	-	-	-	-
	X	72.5	**-26.2**	-18.7	-15.4	-24.1	-22.8	-	-	-	-	-	-	-	-19.7	-14.8	-14.5	-15.7	-16.2	-	-	-	-

Table I.1: Zero-shot cross-lingual transfer performance on five tasks (DEP, POS, NER, XNLI, and XQuAD) with mBERT (B) and XLM-R (X). We show the monolingual EN performance and report drops in performance relative to EN for all target languages. Numbers in bold indicate the largest zero-shot performance drops for each task.

NER, we feed MMT's token-level outputs to a CRF classifier, similar to Peters et al. (2017). For XNLI, we apply a simple softmax classifier on the vector of the sequence start token (`[CLS]` for mBERT; `<s>` for XLM-R); for XQuAD, we pool MMT's representations of all subwords and input it to a span classification head – a linear layer computing the start and the end of the answer.

Training and Evaluation Details. We experiment with mBERT *Base cased* and XLM-R *Base*, both with $L = 12$ transformer layers, hidden state size of $H = 768$, and $A = 12$ self-attention heads.

For XNLI, we limit the inputs to $T = 128$ subword tokens and train in batches of 32 instances. For XQuAD, we limit paragraphs to $T = 384$ tokens and questions to $Q = 64$ tokens. We slide over paragraphs with a window of 128 tokens and train in batches of size 12. For XNLI and XQuAD, we search the following hyperparameter grid: learning rate $\lambda \in \{5 \cdot 10^{-5}, 3 \cdot 10^{-5}\}$; training epochs $n \in \{2, 3\}$. For DEP, POS and NER, we fix the number of training epochs to 20. We train in batches of 32 sentences, with maximal length of $T = 512$ subword tokens. We optimize all models with Adam (Kingma and Ba, 2015).

We report DEP performance in terms of Unlabeled Attachment Scores (UAS).[7] For POS, NER, and XNLI we report accuracy, and for XQuAD, we report the Exact Match (EM) score.

I.3.2 Results and Preliminary Discussion

A summary of the zero-shot cross-lingual transfer results, per target language, is provided in Table I.1. As expected, we observe drops in performance for all tasks and all target languages w.r.t. reference EN performance. However, the drops vary greatly across languages. For example, NER (mBERT) drops mere

[7] Using Labeled Attachment Score (LAS) would make differences in annotation schemes between languages a confounding factor and impede our analysis of effects of language proximity and size of the target language corpora.

2.6% for IT, but enormous 32% for AR; XNLI transfer (XLM-R) yields a moderate 6.1% drop for FR, but a large 20% drop for SW, etc.

At first glance, it appears – as suggested in prior work – that the transfer drops primarily correlate with language proximity: they are more pronounced for languages that are more distant from EN (e.g., JA, ZH, AR, TH, SW). While we see no notable exception to this in the three lower-level tasks, language proximity alone does not explain many of the XNLI and XQuAD results. For instance, RU XNLI (for both mBERT and XLM-R) is comparable to that of ZH, and lower than that for HI and UR: this is despite the fact that, as Indo-European languages, RU, HI, and UR are linguistically closer to EN than ZH. Similarly, we observe comparable performance on XQuAD for TH, RU, and ES.

I.3.3 Analysis

For each task, we now analyze the correlations between transfer performance and **a)** several measures of linguistic proximity (i.e., similarity) between languages and **b)** the size of MMT pretraining corpora of each target language.

Language Vectors and Corpora Sizes. For estimates of linguistic similarity, we rely on language vectors from LANG2VEC, which encode various linguistic features from the URIEL database (Littell et al., 2017). We consider the following LANG2VEC vectors: `syntax` (SYN) vectors encode syntactic properties, e.g., if a subject appears before or after a verb; `phonology` (PHON) vectors encode phonological properties such as the consonant-vowel ratio; `inventory` (INV) vectors denote presence or absence of natural classes of sounds (e.g., voiced uvulars); FAM vectors encode memberships in `language families`; and GEO vectors express orthodromic distances for languages w.r.t. fixed points on the Earth's surface. Language proximity is computed as cosine similarity between the languages' corresponding LANG2VEC vectors: each vector type (e.g., SYN) produces one similarity score (i.e., feature). We couple LANG2VEC features with the z-normalized size of the target language corpus used in MMT pretraining (SIZE).[8]

Correlation Analysis. We first correlate individual features with the zero-shot transfer scores for each task and show the results in Table I.2. Quite intuitively, the zero-shot performance for low-level syntactic tasks – POS and DEP – highly correlates with syntactic language similarity (SYN). SYN also correlates well with transfer results for high-level tasks (except with XLM-R results on XQuAD). Somewhat surprisingly, the phonological language similarity (PHON) correlates best with transfer performance with XLM-R, for all tasks except XNLI, and also for mBERT on POS. For both high-level tasks and both MMTs, we observe very high correlations between transfer performance and size of pretraining corpora of the target language (SIZE). In contrast, SIZE exhibits lower correlations for lower-level tasks (DEP, POS, NER). We believe that this reflect the fact that high-level LU tasks rely on rich representations of

[8] For XLM-R, we take reported sizes of language-specific CC-100 portions (Conneau et al., 2020); for mBERT, we work with sizes of language-specific Wikipedias.

		SYN		PHON		INV		FAM		GEO		SIZE	
Task	**Model**	P	S	P	S	P	S	P	S	P	S	P	S
DEP	XLM-R	0.77	0.78	**0.83**	**0.77**	0.46	-0.04	0.68	0.61	0.80	0.81	0.62	0.47
	mBERT	**0.92**	**0.91**	0.79	0.74	0.55	-0.01	0.76	0.62	0.64	0.69	0.79	0.59
POS	XLM-R	0.68	0.79	**0.81**	**0.81**	0.38	0.02	0.58	0.74	0.80	0.73	0.54	0.46
	mBERT	**0.90**	**0.87**	0.86	0.81	0.57	0.02	0.82	0.80	0.66	0.72	0.47	0.39
NER	XLM-R	0.49	0.49	**0.80**	**0.83**	0.27	0.14	0.47	0.55	0.77	0.81	0.37	0.35
	mBERT	0.60	0.74	**0.81**	**0.84**	0.34	-0.04	0.53	0.58	0.59	0.73	0.42	0.38
XNLI	XLM-R	**0.88**	**0.90**	0.29	0.27	0.31	-0.11	0.63	0.54	0.54	0.74	0.70	0.76
	mBERT	**0.87**	0.86	0.21	0.08	0.29	0.04	0.61	0.47	0.55	0.67	0.77	**0.91**
XQuAD	XLM-R	0.69	0.53	**0.85**	**0.81**	0.62	-0.01	**0.81**	0.54	0.43	0.50	**0.81**	0.55
	mBERT	0.84	0.89	0.56	0.48	0.55	0.22	0.79	0.64	0.51	0.55	**0.89**	**0.96**

Table I.2: Correlations between zero-shot transfer performance with mBERT and XLM-R for different downstream tasks with linguistic proximity features (SYN, PHON, INV, FAM and GEO) and pretraining size of target-language corpora (SIZE). Results reported in terms of Pearson (P) and Spearman (S) correlation coefficients.

Task	**Model**	**Selected features**	P	S	MAE
POS	X	PHON (.75); GEO (.25)	0.77	0.75	10.99
	B	SYN (.99)	0.94	0.90	4.60
DEP	X	PHON (.25); SYN (.18) GEO (.57)	0.81	0.89	10.14
	B	SYN(.99)	0.93	0.92	5.77
NER	X	PHON(.99)	0.80	0.88	4.64
	B	PHON(.99)	0.69	0.82	9.45
XNLI	X	SYN (.51); SIZE (.49)	0.84	0.85	2.01
	B	SYN (.35); SIZE (.34), FAM (.31)	0.89	0.90	2.78
XQuAD	X	PHON (.99)	0.95	0.83	2.89
	B	SIZE (.99)	0.89	0.93	4.76

Table I.3: Results of the meta-regression analysis, i.e., predicting zero-shot transfer performance for mBERT (B) and XLM-R (X). For each task-model pair we list only features with weights ≥ 0.01. P=Pearson; S=Spearman; MAE=Mean Absolute Error.

semantic phenomena of a language, whereas low-level tasks require simpler structural representation of a language – it simply takes more distributional data to acquire the former than the latter.

Meta-Regression. Across the tasks, we observe high correlations between zero-shot transfer results and several features (e.g., SYN, PHON and SIZE). We next test if we can predict the transfer performance for a new language, by (linearly) combining individual features. For each task, we fit a linear regression using transfer results for target languages as labels. With only between 11 and 14 target languages (i.e., instances for fitting the regressor) per task, we resort to leave-one-out cross-validation (LOOCV) to obtain correlations for feature combinations. We perform greedy forward feature selection: in each iteration we add the feature which boosts correlation (obtained via LOOCV) the most; we stop when none of the remaining features further improves the Pearson correlation.

We summarize the results of this meta-regression analysis in Table I.3. For each task-model pair, we list features selected with the greedy feature selection and show (normalized) weights assigned to each feature. Except for NER, combinations of features manage to yield higher correlations with zero-shot transfer results than any of the features on their own. These results empirically confirm our previous intuition that linguistic proximity between the source and target language only partially explains zero-short transfer performance. On XNLI, transfer performance is best explained with the combination of structural similarity between languages (SYN) and the size of the target-language pretraining corpora (SIZE); on XQuAD with mBERT, SIZE alone best explains zero-short transfer scores. Note that the features are mutually quite correlated as well (e.g., languages closer to EN also tend to have larger pretraining corpora): thus if the regressor selects only one feature, this does not mean that other features do not correlate with transfer performance (as shown by Table I.2).

The coefficients in Table I.3 again indicate the importance of SIZE for the language understanding tasks and highlight our core finding: pretraining corpora sizes are stronger features for predicting zero-shot performance in higher-level tasks, whereas the results in lower-level tasks are more affected by typological language proximity.

I.4 From Zero to Hero: Few-Shot

Motivated by the low zero-shot transfer performance for many tasks and languages obtained in §I.3, we now investigate Q4 from §I.1: we aim to mitigate transfer losses with inexpensive few-shot cross-lingual transfer.

Experimental Setup. We rely on the same models, tasks, and evaluation protocols as described in §VIII.3. However, instead of fine-tuning the MMTs on task-specific data in EN only, we continue the fine-tuning process by feeding k additional training examples randomly chosen from reserved target language data portions, disjoint with the test sets.[9] For our low-level tasks, we compare

[9] Note that for XQuAD, we performed the split on the article level to avoid topical overlap. Consequently, for XQuAD k refers to the number of articles.

		k	$k=10$		$k=50$		$k=100$		$k=500$		$k=1000$	
Task	Model	$k=0$	score	Δ	score	Δ	score	Δ	score	Δ	score	Δ
DEP	mBERT	52.96	66.69	13.73	72.67	19.70	74.8	21.84	80.47	27.5	82.74	29.77
	XLM-R	48.60	65.57	16.97	72.19	23.59	74.08	25.48	81.16	32.56	83.33	34.73
POS	mBERT	67.2	80.17	12.96	85.34	18.14	87.09	19.88	91.16	23.96	92.64	25.44
	XLM-R	65.5	80.68	15.18	85.7	20.2	87.59	22.09	91.35	25.85	92.80	27.3
NER	mBERT	79.34	83.18	3.84	84.54	5.20	85.25	5.91	87.9	8.56	89.31	9.97
	XLM-R	85.43	88.06	2.63	91.07	5.64	91.49	6.06	93.69	8.26	93.82	8.39
XNLI	mBERT	65.92	65.89	-0.03	65.08	-0.84	64.92	-1.00	67.41	1.49	68.16	2.24
	XLM-R	73.32	73.73	0.41	73.76	0.45	75.03	1.71	75.34	2.02	75.84	2.52
			$k=2$		$k=4$		$k=6$		$k=8$		$k=10$	
XQUAD	mBERT	45.62	48.12	2.50	48.66	3.04	49.34	3.72	49.91	4.29	50.19	4.57
	XLM-R	53.68	53.73	0.05	53.84	0.17	54.76	1.08	55.56	1.88	55.78	2.10

Table I.4: Results of the few-shot experiments with varying numbers of target-language examples k. For each k, we report performance averaged across languages and the difference (Δ) with respect to the zero-shot setting.

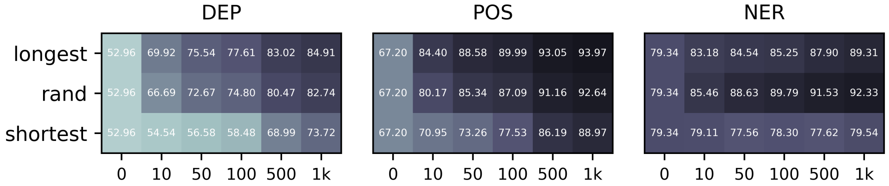


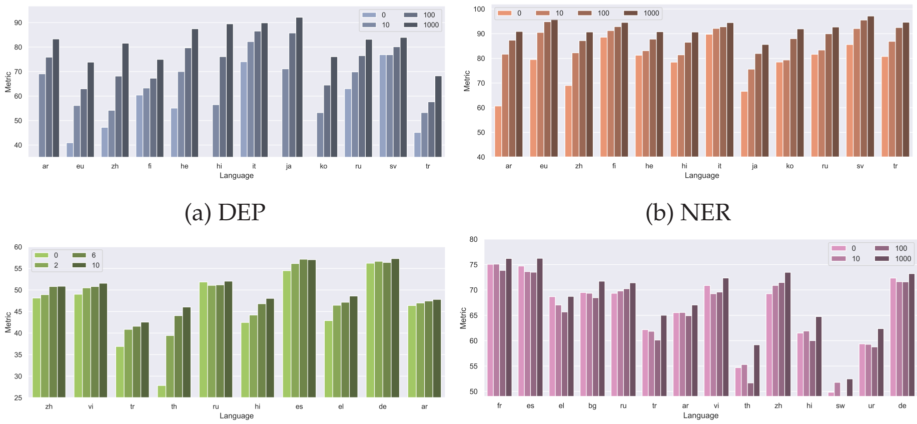
Figure I.1: Heatmap of performance gains for low-level tasks from few-shot transfer with mBERT for different sampling strategies. X-axis: number of target-language instances k; Y-axis: sampling strategy.

three sampling methods: (i) random sampling (RAND) of k target language sentences, (ii) selection of the k shortest (SHORTEST) and (iii) the k longest (LONGEST) sentences.[10] For XNLI and XQuAD, we run the experiments five times and report the average scores.

I.4.1 Results and Discussion

The results on each task, conditioned on the number of examples k and averaged across all target languages, are presented in Table I.4. We note substantial improvements in few-shot learning setups for all tasks. However, the results also reveal notable differences between different types of tasks. For higher-level language understanding tasks the improvements are less pronounced; the maximum gains for XNLI and XQuAD after seeing $k = 1,000$ target-language

[10] In all three cases, we only choose between sentences with ≥ 3 and ≤ 50 tokens.



(a) DEP

(b) NER

(c) XQuAD

(d) XNLI

Figure I.2: Few-shot transfer results with mBERT for each language with varying k for two low-level tasks: a) DEP, b) NER, and two higher-level tasks: c) XQuAD, d) XNLI. For DEP, NER, and XNLI k denotes the number of sampled sentences, for XQuAD, the number of sampled articles.

instances and 10 articles, respectively, are between 2.52 (XLM-R) and 4.57 points (mBERT). On the other hand, the average gains for the lower-level tasks are massive: between 10 (NER) and 30 (DEP) points for mBERT and 8 (NER) and 35 (DEP) points for XLM-R. Moreover, the gains in all lower-level tasks are substantial even when we add only 10 annotated sentences in the target language (on average, up to 17 points on DEP, and 15 points on POS). What is more, our additional experiments (omitted for brevity) show substantial gains for DEP and POS even with fewer than 5 annotated target language sentences. A comparison of different sampling strategies for the lower-level tasks is shown in Figure VIII.6 for mBERT.[11] For DEP and POS, the pattern is very clear and quite expected – adding longer sentences results in better scores. For NER, however, random sampling (RAND) appears to perform best: we hypothesize that this is because: (i) very long sentences are relatively sparse with named entities, resulting in our model seeing mostly negative examples; (ii) shorter sentences contribute less than for DEP and POS because they typically consist of (confirmed by manual inspection) a single named entity mention, without any non-NE tokens.

Figure I.2 illustrates few-shot performance for individual languages on two lower-level (DEP, NER) and two higher-level tasks (XNLI, XQuAD), for different values of k.[12] Across languages, we see a clear trend – more distant target languages benefit much more from the few-shot data. Observe, e.g., SV for DEP or DE for XQuAD. Both are closely related to EN, exhibit high zero-shot transfer

[11] A similar analysis for XLM-R is in the supplementary.

[12] We show per-language scores for POS with mBERT, and all tasks with XLM-R in the Appendix.

performance, and benefit only marginally from few in-language instances. We hypothesize that for such closely related languages, with enough pretraining data, MMT is able to extrapolate the missing language-specific knowledge from few in-language examples; its priors for languages close to EN are already quite sensible and *a priori* offer less room for improvements. In stark contrast, KO (DEP, a) and TH (XQuAD, b), for example, both exhibit poor zero-shot performance and understandably so, given their linguistic distance to EN. Given in-language data, however, both see rapid leaps in performance, displaying gains of almost 40% UAS on DEP (KO), and almost 5% on XQuAD (TH). This can be seen as MMTs' ability to rapidly learn to utilize the multilingual space to adjust its task-specific knowledge for the target language. Other interesting patterns emerge. Particularly interesting are DEP results for JA and AR, where we observe massive UAS improvements with only 10 annotated sentences. For XQuAD, we observe a substantial improvement from only 2 in-language documents for TH. In sum, we see the largest gains from few-shot transfer exactly for languages for which the zero-shot transfer setup yields largest performance drops: languages distant from EN and represented with small corpora in MMT pretraining.

Direct Target Language Few-Shot Fine-Tuning. We have additionally run a set of control experiments in which we bypass the task-specific fine-tuning on the Enhlish data and directly fine-tune the MMTs on the few target language instances. Expectedly, for high-level LU tasks, fine-tuning the MMTs with only a handful of target language examples (i.e., *without* prior fine-tuning in English) yields subpar performance w.r.t. the corresponding model variant that had been previously fine-tuned on English data. For instance, direct few-shot target language fine-tuning of mBERT yields the average XNLI performance of 33.95 for $k = 100$ and 40.19 for $k = 1,000$, respectively (compared to 64.92 and 68.16, respectively, when prior fine-tuning on English data is performed). These findings suggest that fine-tuning with abundant (English) in-task data plus fine-tuning with scarce in-language in-task data yields a truly synergistic effect for higher-level language understanding tasks: the small number of examples in the target language is not sufficient to adapt the MMT directly, but they can provide a substantial edge over fine-tuning only on the English data (i.e., zero-shot transfer).

Somewhat surprisingly, however, for the simpler lower-level tasks, omitting task-specific fine-tuning on the English data and fine-tuning only on few target language instances does not lead to the major deterioration of performance (in fact, in some cases, omitting to fine-tune the MMTs on English data even slightly improves the results): for NER (mBERT) we obtain the average performance of 82.89 and 89.76 for $k = 100$ and $k = 1,000$ respectively, compared to 85.25 and 89.31 obtained respectively with prior English fine-tuning; for POS, the direct few-shot target language fine-tuning yields 87.08 ($k = 100$) and 92.64 ($k = 1,000$). We observe the same trends for the remaining tasks and with XLM-R. This suggests that MMTs can be fine-tuned for lower-level (i.e., simpler) tasks with only a handful of instances.

Task	#inst.	Cost est.	Δ mBERT	Δ XLM-R
POS	1K sents	$73	+25.4	+27.3
DEP	1K sents	$280	+29.8	+34.7
NER	1K sents	$60	+10	+8.4
NLI	1K sent. pairs	$10	+2.24	+2.54
QA	10 docs	$30	+4.5	+2.1

Table I.5: Conversion rates between target language annotation costs and corresponding average performance gains from MMT-based few-shot language transfer.

I.4.2 Cost of Language Transfer Gains

As shown in §I.4.1, moving to few-shot transfer can massively improve performance and reduce the gaps observed with zero-shot transfer, especially for low-resource languages. While additional fine-tuning on few target-language examples is computationally cheap, data annotation may be expensive, especially for minor languages. What are the annotation costs, and how do they translate into performance gains? Table I.5 provides ballpark estimates for our five evaluation tasks; the estimates are based on annotation costs from the literature (Bontcheva et al., 2017; Hovy et al., 2014; Marelli et al., 2014; Rajpurkar et al., 2016; Tratz, 2019). We explain these cost-to-gain conversion estimates in more detail in Appendix I.C).

A provocative high-level question that calls for further discussion in future work can be framed as: are GPU hours effectively more costly[13] than data annotations are in the long run? While MMTs are extremely useful as general-purpose models of language, their potential for some (target) languages can be quickly unlocked by pairing them with a small number of annotated target-language examples. Effectively, this suggests leveraging the best of both worlds, i.e., coupling knowledge encoded in large MMTs with a small annotation effort.

I.5 Conclusion

Research on zero-shot language transfer in NLP is motivated by inherent data scarcity: the fact that most languages have no annotated data for most NLP tasks. Massively multilingual transformers (MMTs) have recently been praised for their zero-shot transfer capabilities that mitigate the data scarcity issue. In this work, we have demonstrated that, similar to earlier language transfer paradigms, MMTs perform poorly in zero-shot transfer to distant target languages, and to languages with smaller monolingual corpora available for exploitation in MMT pretraining. We have presented a detailed empirical analysis of factors affecting zero-shot transfer performance of MMTs across diverse tasks and languages. Our results have revealed that structural language

[13] Financially, but also ecologically (Strubell et al., 2019).

similarity determines the transfer success for lower-level tasks like POS-tagging and dependency parsing; on the other hand, the pretraining corpora size of the target language is crucial for explaining transfer results for higher-level language understanding tasks, such as question answering and natural language inference.

Finally and most importantly, we have shown that the MMT potential on distant and low-resource target languages can be quickly unlocked if they are provided a handful of annotated instances in the target language. This finding provides a strong incentive for intensifying future research efforts that focus on cheap or naturally occurring supervision (Artetxe et al., 2020a; Marchisio et al., 2020; Vulić et al., 2019), quick and simple annotation procedure, and the more effective few-shot transfer learning setups.

Acknowledgements

The work of Anne Lauscher and Goran Glavaš has been supported by the Eliteprogramm of the Baden-Württemberg Stiftung (Grant “AGREE: Algebraic Reasoning over Events from Text and External Knowledge”). The work of Ivan Vulić is supported by the ERC Consolidator Grant LEXICAL (no 648909). We thank the anonymous reviewers for their suggestions and insightful comments.

References

Ammar, W. et al. (2016). “Many Languages, One Parser”. In: *Transactions of the Association for computational Linguistics* vol. 4.

Artetxe, M. and Schwenk, H. (2019). “Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond”. In: *Transactions of the Association for Computational Linguistics* vol. 7.

Artetxe, M. et al. (2017). “Learning bilingual word embeddings with (almost) no bilingual data”. In: *Proceedings of ACL*.

Artetxe, M. et al. (2020a). “A Call for More Rigor in Unsupervised Cross-lingual Learning”. In: *Proceedings of ACL*.

Artetxe, M. et al. (2020b). “On the cross-lingual transferability of monolingual representations”. In: *Proceedings of ACL*.

Artetxe, M. et al. (2020c). “Translation Artifacts in Cross-lingual Transfer Learning”. In: *CoRR* vol. abs/2004.04721.

Bender, E. M. (2011). “On achieving and evaluating language-independence in NLP”. In: *Linguistic Issues in Language Technology* vol. 6, no. 3.

Bontcheva, K. et al. (2017). “Crowdsourcing Named Entity Recognition and Entity Linking Corpora”. en. In: *Handbook of Linguistic Annotation*. Ed. by Ide, N. and Pustejovsky, J.

Camacho-Collados, J. et al. (2016). “Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities”. In: *Artificial Intelligence* vol. 240.

Cao, S. et al. (2020). "Multilingual alignment of contextual word representations". In: *Proceedings of ICLR*.

Conneau, A. et al. (2018). "XNLI: Evaluating Cross-lingual Sentence Representations". In: *Proceedings of EMNLP*.

Conneau, A. et al. (2020). "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.

Dandapat, S. et al. (2009). "Complex linguistic annotation—no easy way out!: A case from Bangla and Hindi POS labeling tasks". In: *Proceedings of the 3rd Linguistic Annotation Workshop*.

Devlin, J. et al. (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)*.

Dozat, T. and Manning, C. D. (2017). "Deep Biaffine Attention for Neural Dependency Parsing". In: *Proceedings of ICLR*.

Eger, S. et al. (2018). "Cross-lingual Argumentation Mining: Machine Translation (and a bit of Projection) is All You Need!" In: *Proceedings of COLING*.

Fort, K. (2016). *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. John Wiley & Sons.

Garrette, D. and Baldridge, J. (2013). "Learning a Part-of-Speech Tagger from Two Hours of Annotation". In: *Proceedings of NAACL-HLT*.

Glavaš, G. and Vulić, I. (2020). "Is Supervised Syntactic Parsing Beneficial for Language Understanding? An Empirical Investigation". In: *CoRR* vol. abs/2008.06788.

Glavaš, G. et al. (2019). "How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions". In: *Proceedings of ACL*.

Hovy, D. et al. (2014). "Experiments with crowdsourced re-annotation of a POS tagging data set". In: *Proceedings of ACL*.

Hu, J. et al. (2020). "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation". In: Proceedings of Machine Learning Research vol. 119. Ed. by III, H. D. and Singh, A.

Joshi, P. et al. (2020). "The State and Fate of Linguistic Diversity and Inclusion in the NLP World". In: *Proceedings of ACL*.

Karthikeyan, K. et al. (2020). "Cross-lingual ability of multilingual BERT: An empirical study". In: *Proceedings of ICLR*.

Kingma, D. P. and Ba, J. (2015). "Adam: A method for stochastic optimization". In: *Proceedings of ICLR*.

Kondratyuk, D. and Straka, M. (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)*.

Lample, G. and Conneau, A. (2019). "Cross-lingual language model pretraining". In: *arXiv preprint arXiv:1901.07291*.

Lehmann, J. et al. (2015). “DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia”. In: *Semantic Web* vol. 6, no. 2.

Libovickỳ, J. et al. (2020). “On the Language Neutrality of Pre-trained Multilingual Representations”. In: *arXiv preprint arXiv:2004.05160*.

Lin, Y.-H. et al. (2019). “Choosing Transfer Languages for Cross-Lingual Learning”. In: *Proceedings of ACL*.

Littell, P. et al. (2017). “URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors”. In: *Proceedings of EACL*.

Liu, Y. et al. (2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv:1907.11692*.

Marchisio, K. et al. (2020). “When Does Unsupervised Machine Translation Work?” In: *CoRR* vol. abs/2004.05516.

Marelli, M. et al. (2014). “A SICK cure for the evaluation of compositional distributional semantic models”. In: *Proceedings of LREC*.

Mayhew, S. et al. (2017). “Cheap translation for cross-lingual named entity recognition”. In: *Proceedings of EMNLP*.

Mikolov, T. et al. (2013). “Exploiting similarities among languages for machine translation”. In: *CoRR* vol. abs/1309.4168. arXiv: 1309.4168.

Moro, A. et al. (2014). “Entity linking meets word sense disambiguation: a unified approach”. In: *Transactions of the Association for Computational Linguistics* vol. 2.

Mrkšić, N. et al. (2017). “Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints”. In: *Transactions of the Association for Computational Linguistics* vol. 5.

Navigli, R. and Ponzetto, S. P. (2012). “BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network”. In: *Artificial Intelligence* vol. 193.

Nivre, J. et al. (2017). *Universal Dependencies 2.1*.

Peters, M. et al. (2017). “Semi-supervised sequence tagging with bidirectional language models”. In: *Proceedings of ACL*.

Pfeiffer, J. et al. (2020). “MAD-X: An Adapter-based Framework for Multi-task Cross-lingual Transfer”. In: *Proceedings of EMNLP*.

Pires, T. et al. (2019). “How multilingual is Multilingual BERT?” In: *arXiv:1906.01502 [cs]*. arXiv: 1906.01502.

Ponti, E. M. et al. (2019). “Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing”. In: *Computational Linguistics* vol. 45, no. 3.

Pruksachatkun, Y. et al. (2020). “Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work?” In: *Proceedings of ACL*.

Rahimi, A. et al. (2019). “Massively Multilingual Transfer for NER”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Raiman, J. R. and Raiman, O. M. (2018). “DeepType: Multilingual entity linking by neural type system evolution”. In: *Proceedings of AAAI*.

Rajpurkar, P. et al. (2016). “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of EMNLP*.

Ruder, S. et al. (2019). “A survey of cross-lingual embedding models”. In: *Journal of Artificial Intelligence Research* vol. 65.

Rönnqvist, S. et al. (2019). “Is Multilingual BERT Fluent in Language Generation?” In: *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*.

Sabou, M. et al. (2012). “Crowdsourcing research opportunities: Lessons from Natural Language Processing”. In: *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*.

Sennrich, R. et al. (2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of ACL*.

Smith, S. L. et al. (2017). “Offline bilingual word vectors, orthogonal transformations and the inverted softmax”. In: *Proceedings of ICLR*.

Strubell, E. et al. (2019). “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of ACL*.

Søgaard, A. et al. (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.

Tratz, S. (2019). “Dependency Tree Annotation with Mechanical Turk”. en. In: *Proceedings of the First Workshop on Aggregating and Analysing Crowdsourced Annotations for NLP*.

Vaswani, A. et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems*.

Vulić, I. et al. (2019). “Do We Really Need Fully Unsupervised Cross-Lingual Embeddings?” 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.

Vulić, I. et al. (2020). “Are All Good Word Vector Spaces Isomorphic?” In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Wenzek, G. et al. (2019). “CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data”. In: *CoRR* vol. abs/1911.00359. arXiv: 1911.00359.

Williams, A. et al. (2018). “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference”. In: *Proceedings of NAACL-HLT*.

Wu, S. and Dredze, M. (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.

Wu, S. et al. (2020). “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.

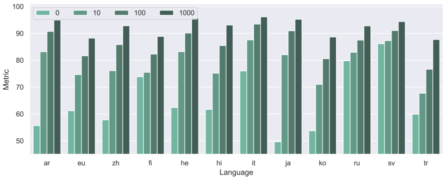
Wu, Y. et al. (2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *CoRR* vol. abs/1609.08144. arXiv: 1609.08144.

Codebase	MMT	Vocab	Params	URL
Allen NLP	–	–	–	https://github.com/allenai/allennlp
HF Trans.	–	–	–	https://github.com/huggingface/transformers
	mBERT	119K	125M	https://huggingface.co/bert-base-multilingual-cased
	XLM-R	250K	125M	https://huggingface.co/xlm-roberta-base

Table I.6: Links to codebases and pretrained models used in this work. For low-level tasks (DEP, POS, NER), we carried out our experiments using the AllenNLP library. For high-level tasks (XNLI, XQuAD), we built our models directly on top of the HuggingFace (HF) Transformers library.

Task	Dataset	URL
Dependency Parsing	UD	https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3105
POS Tagging	UPOS	https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3105
Named Entity Recognition	WikiAnn	https://elisa-ie.github.io/wikiann/
Natural Language Inference	XNLI	https://github.com/facebookresearch/XNLI
Question Answering	XQuAD	https://github.com/deepmind/xquad

Table I.7: Links to the datasets used in our work.



(a) POS

Figure I.3: Graphical illustration of few-shot transfer gains for each language with mBERT, for the remaining task not covered in the main paper: POS.

Appendix I.A Reproducibility

We first provide details on where to obtain datasets and code used in this work.

Code and Dependencies. Our code can be obtained from https://www.dropbox.com/s/o5cxyy92re48xmu/zerohero_code.zip?dl=0. The code is separated in two parts: for experiments related to low-level tasks (DEP, POS, NER) the code is based on the AllenNLP framework; for the experiments on high-level tasks (XNLI, XQuAD), our code directly builds on top of the HuggingFace Transformers framework (Wolf et al., 2019). We provide links to code dependencies and pretrained models in Table I.6.

Datasets. Table I.7 provide links to all datasets that we used in our study, for each of the five tasks (low-level tasks: DEP, POS, NER; high-level tasks: XNLI, XQuAD).

POS	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	55.65	61.19	57.8	73.85	62.38	61.7	76.02	49.65	53.75	79.79	86.15	59.9
10	83.16	74.65	76.1	75.5	83.18	75.19	87.56	82.04	71.02	82.95	87.28	67.73
50	89.18	79.84	83.84	81.4	88.91	83.12	92.04	88.27	77.17	86.07	89.5	74.2
100	90.73	81.63	85.82	82.28	90.12	85.46	93.47	90.95	80.57	87.5	91.06	76.66
500	94.08	86.84	90.78	86.8	94.75	89.69	95.73	94.25	86.48	91.21	93.43	85.29
1000	94.97	88.23	92.83	88.86	95.7	93.09	96.15	95.24	88.64	92.77	94.39	87.72

NER	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	60.69	79.53	69.01	88.59	81.26	78.46	89.77	66.64	78.51	81.64	85.62	80.78
10	81.69	90.51	82.27	91.28	83.12	81.44	92.14	75.64	79.36	83.39	92.09	86.91
50	86.3	93.36	85.6	92.38	87.02	85.04	92.34	78.88	86.94	88.07	95.51	91.93
100	87.37	94.84	87.19	92.88	87.8	86.52	92.79	81.98	88	89.98	95.53	92.5
500	89.74	95.28	89.5	94.01	89.86	89.27	93.8	84.6	90.93	92.18	96.84	94.34
1000	90.92	96.01	90.71	94.57	90.8	90.67	94.5	85.62	91.96	92.71	97.17	94.65

DEP	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	34.72	40.96	47.25	60.44	55.1	33.59	74.05	31.03	35.11	63.03	76.9	45.17
10	69.08	56.16	54.18	63.3	70.02	56.49	82.26	71.12	53.25	69.89	76.88	53.26
50	73.65	61.11	64.39	65.88	78.78	71.48	84.46	82.58	61.11	73.95	79.37	56.78
100	75.91	62.98	68.17	67.31	79.71	76.1	86.53	85.77	64.51	76.51	80.13	57.66
500	81.48	70.33	78.64	71.4	84.81	85.34	89.39	90.38	73.65	81.19	82.87	65.16
1000	83.31	73.85	81.59	74.97	87.47	89.49	89.9	92.18	76.08	83.18	83.95	68.26

XNLI	fr	es	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	de
0	75.05	74.71	68.68	69.50	69.34	62.18	65.53	70.88	54.69	69.26	61.50	49.84	59.38	72.34
10	75.09	73.62	67.04	69.35	69.80	61.86	65.56	69.26	55.30	70.89	61.92	51.79	59.28	71.63
50	74.60	73.91	66.44	68.37	69.05	60.99	64.63	70.29	51.17	71.32	60.08	49.95	58.83	71.43
100	73.85	73.50	65.67	68.47	70.24	60.13	64.93	69.59	51.68	71.46	60.01	48.96	58.78	71.60
500	75.36	74.97	68.04	71.03	70.59	63.21	66.71	72.38	58.12	72.81	64.06	52.26	61.15	73.09
1000	76.20	76.24	68.73	71.73	71.41	65.01	67.04	72.35	59.19	73.47	64.75	52.47	62.38	73.21

XQuAD	zh	vi	tr	th	ru	hi	es	el	de	ar
0	48.14	49.02	36.90	27.84	51.86	42.47	54.48	42.90	56.22	46.40
2	48.93	50.50	40.87	39.43	51.07	44.19	56.14	46.46	56.66	46.99
4	49.72	51.38	40.22	41.24	51.33	45.90	56.62	47.25	56.38	46.57
6	50.81	50.81	41.59	44.04	51.20	46.81	57.14	47.16	56.40	47.45
8	51.53	51.29	41.99	45.28	51.29	47.10	57.45	47.95	57.07	48.21
10	50.87	51.57	42.55	46.05	52.05	48.06	57.03	48.60	57.29	47.82

Table I.8: Detailed per-language few-shot language results with mBERT for different number of target-language data instances k. For low-level tasks, we report results with RAND sampling.

Appendix I.B Full Per-Language Few-Shot Results

We show full per-language few-shot transfer results for all five tasks (DEP, POS, NER, XNLI, XSQuAD) for mBERT and XLM-R in Tables I.8 and I.9, respectively. We visually illustrate the gains from few-shot transfer for individual languages, for mBERT (for the POS task not covered in the main paper) in Figure I.3 and for XLM-R (for all five tasks) in Figure I.4. Finally, we show how the few-shot transfer results with XLM-R for lower-level tasks (DEP, POS, NER) depend on the instance sampling strategy (RAND, SHORTEST, LONGEST) in Figure I.5.

POS	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	59.23	64.41	27.06	78.34	68.94	65.63	77.25	19.28	58.98	81.96	85.54	68.61
10	82.72	76.54	68.3	81.04	84.81	77.08	88.44	78.92	70.5	83.95	87.87	72.33
50	89.14	80.19	77.49	84.94	89.13	84.07	92.51	86.94	76.09	87.29	90.8	79.19
100	90.67	83.38	80.83	86.44	90.3	87.23	93.52	88.78	78.91	88.84	91.79	81.65
500	94.36	88.4	86.61	90.23	94.23	91.4	95.7	92.11	84.37	91.87	94.35	87.64
1000	95.29	89.66	88.86	91.87	95.31	94.26	96.18	93.49	86.88	93.19	95.41	89.71

NER	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	67.03	83.58	56.77	90.69	75.05	78.28	89.25	61.46	76	77.87	89.36	85.43
10	75.45	89.81	79.02	91.14	75.1	78.5	90.02	76.45	74.8	84.5	92.01	88.06
50	82.56	91.63	80.81	92.01	80.34	81.23	91.01	78.13	81.8	87.21	94.72	91.07
100	83.37	93.33	82.77	92.77	82.63	83.88	91.23	79.97	83.06	88.01	94.89	91.49
500	86.95	94.82	85.77	93.78	86.09	87.79	92.44	82.38	87.17	91.02	96.33	93.69
1000	88.36	95.24	87.34	94.3	87.4	89.87	93.25	83.45	88.52	91.66	96.78	93.82

DEP	ar	eu	zh	fi	he	hi	it	ja	ko	ru	sv	tr
0	37.46	42.48	6.61	65.33	53.06	32.94	68.54	11.48	36	62.37	75.72	47.83
10	68.37	56.09	45.67	66.97	70.06	51.93	79.32	70.05	49.88	70.14	77.03	54.93
50	74.9	60.92	57.39	71.35	77.95	67.09	83.97	81.64	59.22	73.55	78.72	59.77
100	77.15	63.46	60.33	71.65	78.27	73.2	84.63	84.3	61.37	75.03	81.52	60.06
500	83.29	72.37	71.52	77.22	86.21	87.06	88.82	88.83	73.1	80.41	85.38	68.88
1000	84.99	75.25	76.2	80.46	88.48	90.81	90.14	90.28	75.35	82.88	85.68	70.68

XNLI	fr	es	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	de	
0	84.25	78.16	78.44	75.39	77.68	75.25	72.99	71.28	74.59	72	73.21	70.02	64.03	66.93	76.45
10	84.26	77.96	78.67	75.77	78.11	76.32	73.31	71.75	75.17	73.18	74.53	69.23	64.09	68.32	77.32
50	84.39	78.69	79.81	76.13	77.57	76.16	73.96	71.2	75.01	71.74	74.47	69.84	61.98	68.06	77.6
100	83.64	79.37	78.87	76.28	77.58	77.42	73.31	71.4	74.83	71.94	74.1	70.54	61.55	67.63	77.84
200	81.57	79.29	79.84	77.01	78.94	77.54	74.81	73.22	76.52	73.91	76.37	71.54	64	68.98	78.42
500	82.69	79.65	79.95	77.34	79.09	77.78	74.08	73.6	77.22	74.32	77.03	71.75	65.37	68.85	78.71
1000	83.74	79.91	80.29	77.39	79.39	77.8	74.92	74.26	77.34	74.8	77.26	72.83	66.77	69.84	78.91

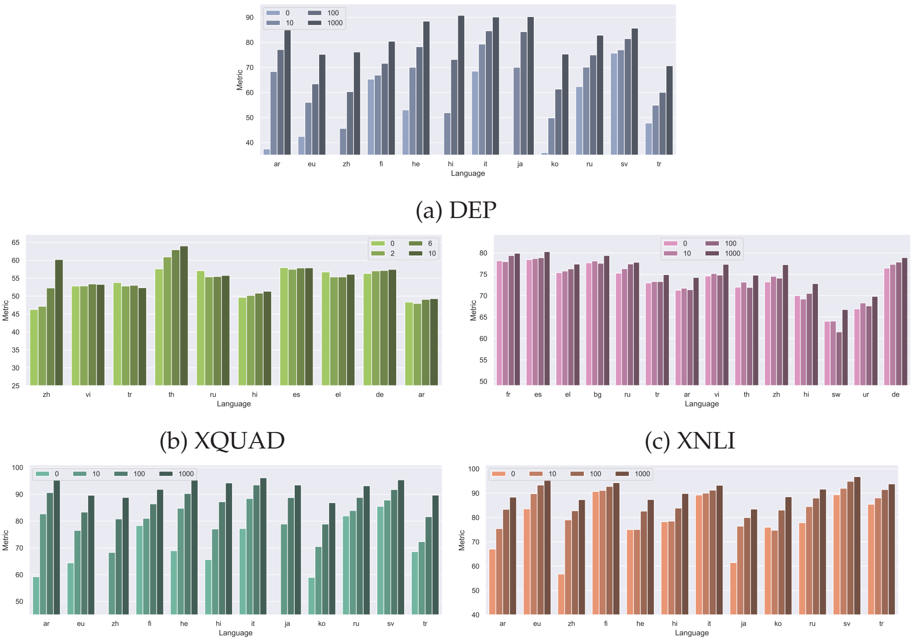
XQuAD	zh	vi	tr	th	ru	hi	es	el	de	ar
0	46.29	52.84	53.82	57.64	57.10	49.67	57.97	56.77	56.33	48.36
2	47.16	52.86	52.84	60.96	55.39	50.20	57.51	55.37	57.05	47.97
4	48.06	53.43	51.88	61.57	54.21	50.28	57.62	55.68	56.72	49.00
6	52.29	53.41	53.03	62.97	55.48	50.85	57.88	55.37	57.16	49.10
8	57.88	53.49	52.47	63.73	55.87	50.96	58.25	55.83	57.05	50.09
10	60.22	53.28	52.36	64.02	55.79	51.38	57.90	56.11	57.47	49.30

Table I.9: Detailed per-language few-shot language results with XLM-R for different number of target-language data instances k. For low-level tasks, we report results with RAND sampling.

Appendix I.C Few-Shot Transfer: Annotation Costs versus Performance Gains

We now present the more detailed explanations for the conversion between the annotation costs and few-shot transfer performance gains, summarized in Table I.5 in the main paper.

Natural Language Inference. Marelli et al. (2014) reportedly paid $\$2,030$ for 200k judgements, which would amount to $\$0.01015$ per NLI instance and, in turn, to $\$10.15$ for $1,000$ annotations. In our few-shot experiments this would



(a) DEP

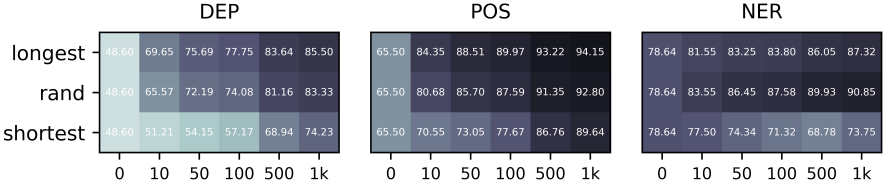
(b) XQUAD

(c) XNLI

(d) POS

(e) NER

Figure I.4: Graphical illustration of few-shot transfer gains for individual languages, for XLM-R and all languages.



DEP

	0	10	50	100	500	1k
longest	48.60	69.65	75.69	77.75	83.64	85.50
rand	48.60	65.57	72.19	74.08	81.16	83.33
shortest	48.60	51.21	54.15	57.17	68.94	74.23

POS

	0	10	50	100	500	1k
longest	65.50	84.35	88.51	89.97	93.22	94.15
rand	65.50	80.68	85.70	87.59	91.35	92.80
shortest	65.50	70.55	73.05	77.67	86.76	89.64

NER

	0	10	50	100	500	1k
longest	78.64	81.55	83.25	83.80	86.05	87.32
rand	78.64	83.55	86.45	87.58	89.93	90.85
shortest	78.64	77.50	74.34	71.32	68.78	73.75

Figure I.5: Heatmap of performance gains for low-level tasks from few-shot transfer with XLM-R for different sampling strategies. X-axis: number of target-language instances k; Y-axis: sampling strategy.

yield an average improvement of 2.24 and 2.52 accuracy points for mBERT and XLM-R, respectively. It is also possible to translate the English data directly via professional translation services as done with the XNLI dataset and XQuAD: the platforms for hiring professionals such as Upwork show that it is possible to find qualified translators even for lower-resource languages: e.g., the translation cost estimate for Zulu is \$12.5-\$16/h, or \$19/h for the Basque language.

Question Answering. Rajpurkar et al. (2016) report a payment cost of \$9 per hour and a time effort of 4 minutes per paragraph. With an average of 5 paragraphs per article, our few-shot scenario (10 articles) roughly requires 50 paragraphs-level annotations, i.e., 200 minutes of annotation effort and would in total cost around \$30 (for respective performance improvements of 4.6 and 2.1 points for mBERT and XLM-R).

On the one hand, compared to language understanding tasks, our lower-level (DEP, POS) tasks are presumably more expensive to annotate, as they require some linguistic knowledge and annotation training. On the other hand, as shown in our few-shot experiments, we typically need much fewer annotated instances (i.e., we observe high gains with already 10 target language sentences) for substantial gains in these tasks.

Dependency Parsing. Tratz (2019) provide an overview of crowd-sourcing annotations for dependency parsing; they report obtaining a fully correct dependency tree from at least one annotator for 72% of sentences. At the reported cost of \$0.28 per sentence this amounts to spending \$280 for annotating $1,000$ sentences. Somewhat shockingly, annotating 10 sentences with dependency trees – which for particular target languages like AR and JA corresponds to performance gains of 30-40 UAS points (see Figure I.2) – amounts to spending merely \$3-5.

Part-of-Speech Tagging. Hovy et al. (2014) measure agreement of crowdsourced POS annotations with expert annotations; they crowdsource annotations for 1,000 tweets, at a cost of \$0.05 for every 10 tokens. With a total of $14,619$ tokens in the corpus, this amounts to approximately \$73 for $1,000$ tweets, which is $\geq 1,000$ sentences.[14] Based on Table I.4, 2 hours of POS annotation work translates to gains of up to 20-22 points on average over zero-shot transfer methods.

Named Entity Recognition. Bontcheva et al. (2017) provide estimates for crowdsourcing annotation for named entity recognition; they pay \$0.06 per sentence, resulting in \$60 cost for $1,000$ annotated sentences. At a median pay of \$11.37/hr, this amounts to around 190 sentences annotated in an hour. In other words, in less than 3 hours, we can collect more than 500 annotated examples. According to Table I.4, this can result in gains of 8+ points on average, and even more for some languages (e.g., 27 points for AR).

[14] Note, however, that lower-level tasks do come with an additional risk of poorer quality annotation, due to crowdsourced annotators not being experts. Garrette and Baldridge (2013) report that even for truly low-resource languages (e.g., Kinyarwanda, Malagasy), it is possible to obtain $\approx$ 100 POS-annotated sentences in 2 hours.

Paper II

Multilingual ELMo and the Effects of Corpus Sampling

Vinit Ravishankar, Andrey Kutuzov, Lilja Øvrelid, Erik Velldal

Appears in *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, May 2021, pp. 378-384.

Abstract

Multilingual pretrained language models are rapidly gaining popularity in NLP systems for non-English languages. Most of these models feature an important corpus sampling step in the process of accumulating training data in different languages, to ensure that the signal from better resourced languages does not drown out poorly resourced ones. In this study, we train multiple multilingual recurrent language models, based on the ELMo architecture, and analyse both the effect of varying corpus size ratios on downstream performance, as well as the performance difference between monolingual models for each language, and broader multilingual language models. As part of this effort, we also make these trained models available for public use.

Contents

II.1 Introduction

As part of the recent emphasis on language model pretraining, there also has been considerable focus on multilingual language model pretraining; this is distinguished from merely training language models in multiple languages by the creation of a multilingual space. These have proved to be very useful in ‘zero-shot learning’; i.e., training on a well-resourced language (typically

English), and relying on the encoder's multilingual space to create reasonable priors across languages.

The main motivation of this paper is to study the effect of corpus sampling strategy on downstream performance. Further, we also examine the utility of multilingual models (when constrained to monolingual tasks), over individual monolingual models, one per language. This paper therefore has two main contributions: the first of these is a multilingual ELMo model that we hope would see further use in probing studies as well as evaluative studies, downstream; we train these models over 13 languages, namely Arabic, Basque, Chinese, English, Finnish, Hebrew, Hindi, Italian, Japanese, Korean, Russian, Swedish and Turkish. The second contribution is an analysis of sampling mechanism on downstream performance; we elaborate on this later.

In Section II.2 of this paper, we contextualise our work in the present literature. Section II.3 describes our experimental setup and Section II.4 our results. Finally, we conclude with a discussion of our results in Section II.5.

II.2 Prior work

Multilingual embedding architectures (static or contextualised) are different from cross-lingual ones Liu et al., 2019; Ruder et al., 2019 in that they are not products of aligning several monolingual models. Instead, a deep neural model is trained end to end on texts in multiple languages, thus making the whole process more straightforward and yielding truly multilingual representations Pires et al., 2019. Following Artetxe et al. (2020), we will use the term 'deep multilingual pretraining' for such approaches.

One of the early examples of deep multilingual pretraining was BERT, which featured a multilingual variant trained on the 104 largest language-specific Wikipedias Devlin et al., 2019. To counter the effects of some languages having overwhelmingly larger Wikipedias than others, Devlin et al. (2019) used exponentially smoothed data weighting; i.e., they exponentiated the probability of a token being in a certain language by a certain α, and re-normalised. This has the effect of 'squashing' the distribution of languages in their training data; larger languages become smaller, to avoid drowning out the signal from smaller languages. One can also look at this technique as a sort of sampling. Other multilingual models, such as XLM Lample and Conneau, 2019 and its larger variant, XLM-R Conneau et al., 2020, use different values of α for this sampling (0.5 and 0.3 respectively). The current paper is aimed at analysing the effects of different α choices; in spirit, this work is very similar to Arivazhagan et al. (2019); where it differs is our analysis on downstream tasks, as opposed to machine translation, where models are trained and evaluated on a very specific task. We also position our work as a resource, and we make our multilingual ELMo models available for public use.

II.3 Experimental setup

II.3.1 Background

When taken to its logical extreme, sampling essentially reduces to truncation, where all languages have the same amount of data; thus, in theory, in a truncated model, no language ought to dominate any other. Of course, for much larger models, like the 104-language BERT, this is unfeasible, as the smallest languages are too small to create meaningful models. By selecting a set of languages such that the smallest language is still reasonably sized for the language model being trained, however, we hope to experimentally determine whether truncation leads to truly neutral, equally capable multilingual spaces; if not, we attempt to answer the question of whether compression helps at all.

Our encoder of choice for this analysis is an LSTM-based ELMo architecture introduced by Peters et al. (2018). This might strike some as a curious choice of model, given the (now) much wider use of transformer-based architectures. There are several factors that make ELMo more suitable for our analysis. Our main motivation was, of course, resources – ELMo is far cheaper to train, computationally. Next, while pre-trained ELMo models already exist for several languages Che et al., 2018; Ulčar and Robnik-Šikonja, 2020, there is, to the best of our knowledge, no multilingual ELMo. The release of our multilingual model may therefore also prove to be useful in the domain of probing, encouraging research on multilingual encoders, constrained to recurrent encoders.

II.3.2 Sampling

Our initial starting point for collecting the language model training corpora were the CoNLL 2017 Wikipedia/Common Crawl dumps released as part of the shared task on Universal Dependencies parsing (Ginter et al., 2017); we extracted the Wikipedia portions of these corpora for our set of 13 languages. This gives us a set of fairly typologically distinct languages, that still are not entirely poorly resourced. The smallest language in this collection, Hindi, has ∼ 91M tokens, which we deemed sufficient to train a reasonable ELMo model.

Despite eliminating Common Crawl data, this gave us, for our set of languages, a total corpus size of approximately 35B tokens, which would be an unfeasible amount of data given computational constraints. We therefore selected a baseline model to be somewhat synthetic – note that this is a perfectly valid choice given our goals, which were to compare various sampling exponents. Our 'default' model, therefore, was trained on data that we obtained by weighting this 'real-world' Wikipedia data. The largest α we could use, that would still allow for feasible training, was $\alpha = 0.4$ (further on, we refer to this model as M0.4); this gave us a total corpus size of ∼4B tokens. Our second, relatively more compressed model, used $\alpha = 0.2$ (further on, M0.2); giving us a total corpus size of ∼2BIL tokens; for our final, most compressed model (further on, TRUNC), we merely truncated each corpus to the size of our smallest corpus (Hindi; 91M), giving us a corpus sized ∼1.2B tokens. Sampling was carried out as follows: if the probability of a token being sampled from a certain language i is p_i, the adjusted probability is given by $q_i = \frac{p_i}{\sum_{j=1}^{N} p_j}$. Note that this is a similar sampling strategy to the one followed by more popular models, like mBERT. We trained an out-of-the box ELMo encoder for approximately the same number of steps on each corpus; this was equivalent to 2 epochs for M0.4 and 3 for M0.2.

Detailed training hyperparameters and precise corpus sizes are presented in Appendices A and B.

II.3.3 Tasks

While there is a dizzying array of downstream evaluation tasks for monolingual models, looking to evaluate multilingual models is a bit harder. We settled on a range of tasks in two different groups:

1. **Monolingual tasks**: these tasks directly test the monolingual capabilities of the model, per language. We include PoS tagging and dependency parsing in this category. In addition to our multilingual models, we also evaluate our monolingual ELMo variants on these tasks.

2. **Transfer tasks**: these tasks involve leveraging the model's multilingual space, to transfer knowledge from the language it was trained on, to the language it is being evaluated on. These tasks include natural language inference and text retrieval; we also convert PoS tagging into a transfer task, by training our model on English and asking it to tag text in other languages.

In an attempt to illuminate precisely what the contribution of the different ELMo models is, we ensure that our decoder architectures – that translate from ELMo's representations to the task's label space – are kept relatively simple, particularly for lower-level tasks. We freeze ELMo's parameters: this is not a study on fine-tuning.

The tasks that we select are a subset of the tasks mentioned in XTREME (Hu et al., 2020); i.e., the subset most suitable to the languages we trained our encoder on. A brief description follows:

PoS tagging: For part-of-speech tagging, we use Universal Dependencies part-of-speech tagged corpora (Nivre et al., 2020). Built on top of our ELMo-encoder is a simple MLP, that maps representations onto the PoS label space.

PoS tagging (transfer): We use the same architecture as for regular PoS tagging, but train on English and evaluate on our target languages.

Dependency parsing: We use dependency-annotated Universal Dependencies corpora; our metrics are both unlabelled and labelled attachment scores (UAS/LAS). Our parsing architecture is a biaffine graph-based parser (Dozat and Manning, 2018).

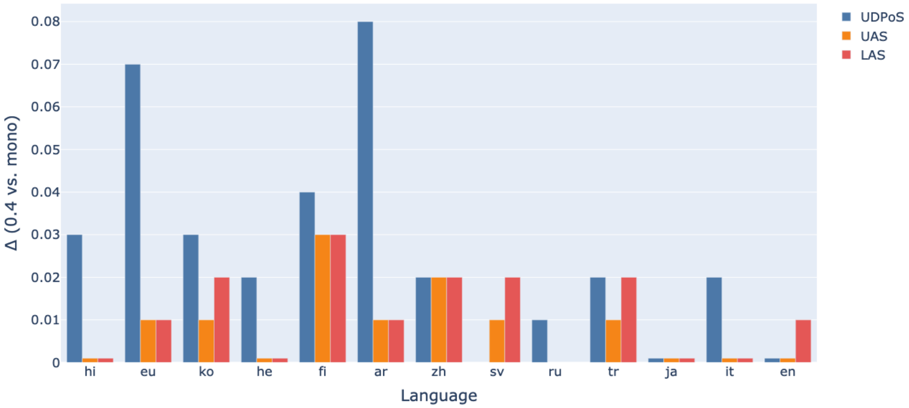


Figure II.1: Performance difference between monolingual and multilingual models, on our monolingual tasks. Absent bars indicate that the language was missing.

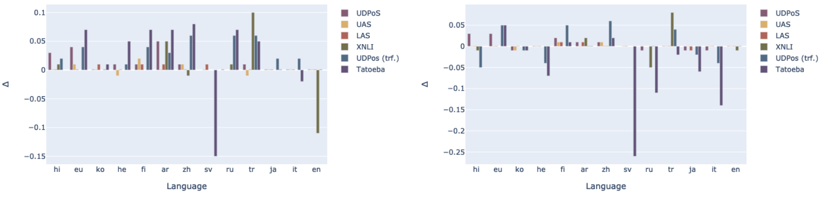
XNLI: A transfer-based language inference task; we use Chen et al.'s 2017 ESIM architecture, train a tagging head on English, and evaluate on the translated dev portions of other languages (Conneau et al., 2018).

Tatoeba: The task here is to pick out, for each sentence in our source corpus (English), the appropriate translation of the sentence in our target language corpus. This, in a sense, is the most 'raw' tasks; target language sentences are ranked based on similarity. We follow Hu et al. (2020) and use the Tatoeba dataset.

We tokenize all our text using the relevant UDPipe (Straka and Straková, 2017) model, and train/evaluate on each task three times; the scores we report are mean scores.

II.4 Results

First, we examine the costs of multilingualism, as far as monolingual tasks are concerned. We present our results on our monolingual tasks in Figure II.1. Monolingual models appear to perform consistently better, particularly PoS tagging; this appears to be especially true for our under-resourced languages, strengthening the claim that compression is necessary to avoid drowning out signal. For PoS tagging, the correlation between performance difference (monolingual vs. M0.4) and corpus size is highly significant ($\rho = 0.74; p = 0.006$).



(a) M0.2 vs. M0.4

(b) TRUNC vs. M0.4

Figure II.2: Performance differences between our models on our selected tasks.

	PoS	UAS	LAS	PoS (trf.)	XNLI	Tatoeba
MONO	0.86	0.86	0.81	-	-	-
M0.4	0.83	0.85	0.80	0.36	0.45	0.18
M0.2	**0.84**	0.85	0.80	**0.39**	**0.46**	**0.21**
TRUNC	0.83	0.85	0.80	0.36	0.45	0.13

Table II.1: Average scores for each task and encoder; non-monolingual best scores in bold.

We find that compression appears to result in visible improvements, when moving from $\alpha = 0.4$ to $\alpha = 0.2$. These improvements, while not dramatic, apply across the board (see Table II.1), over virtually all task/language combinations; this is visible in Figure II.2a. Note the drop in performance on certain tasks for English, Swedish and Italian – we hypothesise that this is due to Swedish and Italian being closer to English (our most-sampled language), and therefore suffering from the combination of the drop in their corpus sizes, as well as the more significant drop in English corpus size. The Pearson correlation between the trend in performance for PoS tagging and the size of a language's corpus is statistically significant ($\rho = 0.65; p = 0.02$); note that while this is over multiple points, it is single runs per data point.

Figure II.2b also shows the difference in performance between the truncated model, TRUNC, and M0.4; this is a lot less convincing than the difference to M0.2, indicating that no additional advantage is to be gained by downsampling data for better-resourced languages.

We include full, detailed results in Appendix C.

Cross-lingual differences Finally, in an attempt to study the differences in model performance across languages, we examine the results of all models on Tatoeba. This task has numerous advantages for a more detailed analysis; i) it covers all our languages, bar Hindi, ii) the results have significant variance across languages, and iii) the task does not involve any additional training. We present these results in Figure II.3.

We observe that M0.2 consistently appears to perform better, as illustrated earlier. Performance does not appear to have much correlation with corpus

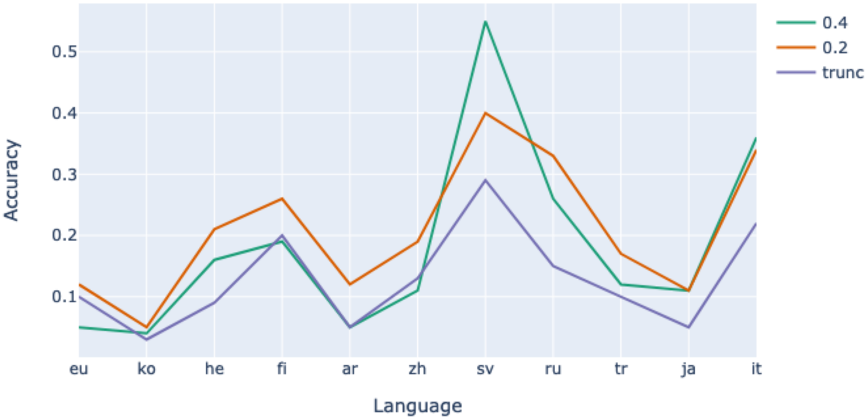


Figure II.3: Accuracy on Tatoeba per model

size; however, the languages for which M0.4 performs better are Swedish and Italian, coincidentally, the only other Latin-scripted Indo-European languages. Given the specific nature of Tatoeba, which involves picking out appropriate translations, these results make more sense: these languages receive not only the advantage of having more data for themselves, but also from the additional data available to English, which in turn optimises their biases solely by virtue of language similarity.

II.5 Discussion

Our results allow us to draw conclusions that come across as very 'safe': some compression helps, too much hurts; when compression does help, however, the margin appears rather moderate yet significant for most tasks, even given fewer training cycles. Immediately visible differences along linguistic lines do not emerge when ratios differ, despite the relative linguistic diversity of our language choices; we defer analysis of this to a future work, that is less focused on downstream analysis, and more on carefully designed probes that might illuminate the difference between our models' internal spaces. Note that a possible confounding factor in our results is also the complexity of the architectures we build on top of mELMO: they also have significant learning capacity, and it is not implausible that whatever differences there are between our models, are drowned out by highly parameterised downstream decoders.

To reiterate, this study is not (nor does it aim to be) a replication of models with far larger parameter spaces and more training data. This is something of a middle-of-the-road approach; future work could involve this sort of evaluation on downscaled transformer models, which we shy away from in order to provide a usable model release. We hope that the differences between these models provide some insight, and pave the way for further research, not only specifically addressing the question of sampling from a perspective of performance, but also analytically. There has already been considerable work in this direction on multilingual variants of BERT (Chi et al., 2020; Pires et al., 2019), and we hope that this work motivates papers applying the same to recurrent mELMo, as well as comparing and contrasting the two. The ELMo models described in this paper are publicly released via NLPL Vector Repository.[1]

Acknowledgements

Our experiments were run on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway, under the NeIC-NLPL umbrella.

References

Arivazhagan, N. et al. (2019). "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges". In: *arXiv:1907.05019 [cs]*. arXiv: 1907.05019.

Artetxe, M. et al. (2020). "A Call for More Rigor in Unsupervised Cross-lingual Learning". In: *Proceedings of ACL*.

Che, W. et al. (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*.

Chen, Q. et al. (2017). "Enhanced LSTM for Natural Language Inference". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. arXiv: 1609.06038.

Chi, E. A. et al. (2020). "Finding Universal Grammatical Relations in Multilingual BERT". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Conneau, A. et al. (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.

Conneau, A. et al. (2020). "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.

[1] http://vectors.nlpl.eu/repository/

Devlin, J. et al. (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.

Dozat, T. and Manning, C. D. (2018). “Simpler but More Accurate Semantic Dependency Parsing”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Ginter, F. et al. (2017). *CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Hu, J. et al. (2020). “XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation”. In: Proceedings of Machine Learning Research vol. 119. Ed. by III, H. D. and Singh, A.

Lample, G. and Conneau, A. (2019). “Cross-lingual language model pretraining”. In: *arXiv preprint arXiv:1901.07291*.

Liu, Q. et al. (2019). “Investigating Cross-Lingual Alignment Methods for Contextualized Embeddings with Token-Level Evaluation”. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*.

Nivre, J. et al. (2020). “Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Peters, M. et al. (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.

Pires, T. et al. (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.

Ruder, S. et al. (2019). “A Survey of Cross-lingual Word Embedding Models”. In: *Journal of Artificial Intelligence Research* vol. 65.

Straka, M. and Straková, J. (2017). “Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe”. In: *CoNLL 2017*.

Ulčar, M. and Robnik-Šikonja, M. (2020). “High Quality ELMo Embeddings for Seven Less-Resourced Languages”. In: *Proceedings of the 12th Language Resources and Evaluation Conference*.

Appendix II.A Hyperparameters

Param	**Value**
Layers	2
Output dimensionality	2048
Batch size	192
Negative samples per batch	4096
Vocabulary size	100,000
Number of epochs	2 (M0.4); 3(M0.2)

Table II.2: Models were bidirectional LSTMs. Monolingual models were trained on individual sizes given at $\alpha = 0.4$.

Appendix II.B Corpus sizes

Language	**AR**	**EN**	**EU**	**FI**	**HE**	**HI**	**IT**	**JA**	**KO**	**RU**	**SV**	**TR**	**ZH**	Total
M0.4	242.29	585.52	113.42	239.57	208.46	91.74	468.45	460.53	184.63	379.9	366.86	396.01	282.76	4020.14
M0.2	149.09	231.76	102.01	148.25	138.29	91.74	207.3	205.54	130.15	186.68	183.45	190.6	161.06	2125.92
TRUNC	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	1192.62

Table II.3: Corpus sizes, in million tokens

Appendix II.C Detailed results

Language		AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
POS	MONO	0.89	0.89	0.88	0.82	0.84	0.9	0.91	0.94	0.67	0.88	-	0.83	0.86
	0.4	0.81	0.89	0.81	0.78	0.82	0.87	0.89	0.94	0.64	0.87	-	0.81	0.84
	0.2	0.86	0.89	0.85	0.79	0.83	0.9	0.89	0.94	0.64	0.87	-	0.82	0.85
	TRUNC	0.82	0.89	0.84	0.8	0.82	0.9	0.88	0.93	0.63	0.86	-	0.81	0.85
UAS	MONO	0.86	0.89	0.84	0.88	0.89	0.94	0.93	0.95	0.8	-	0.85	0.69	0.8
	M0.4	0.85	0.89	0.83	0.85	0.89	0.94	0.93	0.95	0.79	-	0.84	0.68	0.78
	M0.2	0.85	0.89	0.84	0.87	0.88	0.94	0.93	0.95	0.79	-	0.84	0.67	0.79
	TRUNC	0.85	0.89	0.83	0.86	0.89	0.94	0.93	0.95	0.78	-	0.84	0.68	0.79
LAS	MONO	0.79	0.86	0.79	0.84	0.84	0.9	0.9	0.94	0.74	-	0.81	0.59	0.74
	0.4	0.78	0.85	0.78	0.81	0.84	0.9	0.9	0.94	0.72	-	0.79	0.57	0.72
	0.2	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.94	0.73	-	0.8	0.57	0.72
	TRUNC	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.93	0.72	-	0.79	0.57	0.72
POS (trf.)	0.4	0.23	0.89	0.25	0.43	0.36	0.31	0.52	0.22	0.18	0.49	-	0.23	0.22
	0.2	0.26	0.89	0.29	0.47	0.37	0.33	0.54	0.24	0.18	0.55	-	0.29	0.28
	TRUNC	0.23	0.89	0.3	0.48	0.32	0.26	0.48	0.2	0.17	0.49	-	0.27	0.28
XNLI	M0.4	0.41	0.67	-	-	-	0.44	-	-	-	0.48	-	0.35	0.35
	M0.2	0.46	0.56	-	-	-	0.45	-	-	-	0.49	-	0.45	0.34
	TRUNC	0.43	0.66	-	-	-	0.43	-	-	-	0.43	-	0.43	0.35
Tatoeba	0.4	0.05	-	0.05	0.19	0.16	-	0.36	0.11	0.04	0.26	0.55	0.12	0.11
	0.2	0.12	-	0.12	0.26	0.21	-	0.34	0.11	0.05	0.33	0.4	0.17	0.19
	TRUNC	0.05	-	0.1	0.2	0.09	-	0.22	0.05	0.03	0.15	0.29	0.1	0.13

Table II.4: Full score table across all languages, tasks and models

Paper III

The Effects of Corpus Choice and Morphosyntax on Multilingual Space Induction

Vinit Ravishankar, Joakim Nivre

Appears in *Findings of the Association for Computational Linguistics: EMNLP 2022*, December 2022, pp. 4130–4139.

Abstract

In an effort to study the inductive biases of language models, numerous studies have attempted to use linguistically motivated tasks as a proxy of sorts, wherein performance on these tasks would imply an inductive bias towards a specific linguistic phenomenon. In this study, we attempt to analyse the inductive biases of language models with respect to natural language phenomena in the context of building multilingual embedding spaces. We sample corpora from 2 sources in 15 languages and train language models on pseudo-bilingual variants of each corpus, created by duplicating each corpus and shifting token indices for half the resulting corpus. We evaluate the cross-lingual capabilities of these LMs, and show that while correlations with language families tend to be weak, other corpus-level characteristics, such as type-token ratio, tend to be more strongly correlated. Finally, we show that multilingual spaces can be built, albeit less effectively, even when additional destructive perturbations are applied to the training corpora, implying that (effectively) bag-of-words models also have an inductive bias that is sufficient for inducing multilingual spaces.

Contents

III.1 Introduction

A variety of proxies and analytical methods have been used to study the inductive biases of language models towards natural language. This work includes targeted syntactic evaluation (Gulordava et al., 2018; Linzen et al., 2016), language model responses to formulaic synthetic languages (Ravfogel et al., 2019; White and Cotterell, 2021), as well as attempts to correlate differences in language modeling performance to language features over a wide range of languages (Cotterell et al., 2018).

In this paper, we combine two strands that have, of late, been fairly active research threads. The first of these concerns the inductive biases of language models towards languages that exhibit a specific grammar; the second addresses the inductive biases of these models towards multilingualism, which in this context refers to a model's ability to build a multilingual space (rather than distinct monolingual spaces), when trained on corpora consisting of text in multiple languages.

Prior work in this domain is focused on either a) quantifying language model performance across a variety of languages, or b) studying the effects of different architectural components on the quality of the induced multilingual space. We attempt to unite the two strands of research by studying transformer-based masked language models in an effort to quantify the extent to which the grammar of the language being modelled affects the model's ability to build a multilingual space. We use Dufter and Schütze's (2021) metrics, namely word translation and sentence retrieval, as a proxy for the utility of this space. Our main findings are:

- Masked language models are capable of building multilingual spaces even when destructive perrturbations, like lemmatisation and shuffling, are applied to the training corpora.
- Multilingual performance is only weakly correlated with languages and language families.
- Multilingual performance correlates better with corpus-level statistics like type-token ratio, and the frequency of *hapax legomena*.

III.2 Related Work

Language modelling There has been a considerable amount of research addressing inductive biases that language models may have towards specific grammatical patterns, or towards natural languages with specific structures. An early study by Cotterell et al. (2018) demonstrates, over 21 languages, that certain languages are harder to model than others; the authors find that model performance correlates with the richness of a language's (inflectional) morphology. Later work by Mielke et al. (2019) shows contradictory findings; the authors extend these experiments to 69 languages and find that morphological complexity does not correlate as strongly with performance as simpler factors like vocabulary size and sentence length do.

Other work involves studying how language modelling is affected by manually altering corpora. Ravfogel et al. (2019) train RNN-based models on English, altered to display different word orders and different degrees of morphological agreement; White and Cotterell (2021) generate corpora of natural language sentences, with constituents permuted based on Boolean switches, and show that recurrent language models show little variance in performance across word orders, compared to transformers.

Multilingualism Moving beyond monolingual language modelling, we examine the numerous works analysing what precisely multilingual language models need, in order to form an adequate multilingual space, which is quantified by measuring a model's performance on some multilingual task. Pires et al. (2019) show that subword overlap tends to improve multilingual alignment, though overlap is by no means necessary, as languages with different scripts can exist in the same multilingual space. Deshpande et al. (2021) show that while structurally similar languages do not necessarily need subword overlap, dissimilar languages rely heavily on overlap; they also show that well-aligned non-contextual word embedding spaces allow for better transfer.

On the other hand, Artetxe et al. (2020) have somewhat contradictory results, and show that neither shared vocabulary items nor joint pre-training are essential to build a multilingual encoder. K et al. (2020) and Dufter and Schütze (2021) analyse encoders from an architectural point of view. The former work shows that model depth (and not the number of attention heads) contributes to transfer performance, even when the number of parameters is kept constant. The latter points out that multilingual spaces exist because languages are forced to share parameters, and that even in the absence of shared subwords and special tokens, position embeddings play a significant role in building these spaces. Dufter and Schütze (2021) go on to show that the removal of shared position embeddings is sufficient to reduce a model's multilingual performance (as measured on word translation and sentence retrieval) to approximately random. This, we show, is not universally the case.

III.3 Methodology

III.3.1 General approach

In order to evaluate the quality of our models' multilingual spaces, we use word translation and sentence retrieval as proxy tasks; this contrasts with, for example, Deshpande et al. (2021), who use (zero-shot) transfer performance instead. We avoid this largely due to performance constraints: small models are unlikely to be parameterised enough to handle transfer.

To create synthetic multilingual (more precisely, bilingual) corpora, we follow the approach of K et al. (2020) and Dufter and Schütze (2021). Starting from a monolingual corpus, we shift the vocabulary index for every token in the original corpus up by the model's vocabulary size. For instance, the token

Default he spent most of his childhood in sunamganj with his mother . david s. mack (born 1941) is an american businessman . he spent most of his childhood in sunamganj with his mother . david s. mack (born 1941) is an american businessman .
Lemmatised the episode be generally well receive . the software be sell and support only in japan . the episode be generally well receive . the software be sell and support only in japan .
Shuffled most his with in of childhood spent sunamganj . mother his he s. american . born is david 1941) businessman an (mack most his with in of childhood spent sunamganj . mother his he s. american . born is david 1941) businessman an (mack
Corrupted be generally . receive well episode the software be the sell in and support . japan only be generally . receive well episode the software be the sell in and support . japan only

Table III.1: Sample sentences extracted from real corpora, with each of our modifications applied. Note that while the original and lemmatised corpora are sampled differently, the shuffled and corrupted corpora are modified variants of the former.

`convenient`, with token index 42, would have a "mirror" `::convenient`, with token index 2090. This effectively gives us a parallel second half, which has the same structure as the original language, but a guarantee of no vocabulary overlap.

While this is a somewhat unrealistic simulation – after all, multilingual models are trained on languages with *different* structures – we use our formulation in order to a) have a simplified test bed where the *structure* of the language plays a role, but the *structural differences* between the two languages are ignored; and b) to avoid the complexity of the experimental space from exploding, when each language can conceivably be paired with every other language.

III.3.2 Data

In an effort to have a reasonably comprehensive search space of languages, we experiment over two corpora (Wikipedia and Common Crawl) and fifteen languages – namely Arabic, Czech, Danish, German, English, Spanish, Finnish, French, Hebrew, Italian, Dutch, Polish, Portuguese, Russian and Swedish. While Indo-European languages are still rather overrepresented in our data, these languages exhibit a wide range of head-depedendent entropies (Levshina, 2019). This is also part of the reason we avoid completely synthetic corpora: while it is trivial to generate synthetic corpora from some descriptive grammar, the *stochasticity* and random variation inherent to most natural languages is harder to synthetically model. Both corpora have been parsed into Universal Dependencies (UD) (Marneffe et al., 2021; Nivre et al., 2016; Nivre et al., 2020).

From each of the large corpora (Wikipedia and Common Crawl), we sample five corpora of 20k sentences for each language, with different random seeds, and split them into train and validation splits of 15k and 5k tokens, respectively. We employ a number of simple heuristics to filter out sentences that we suspect to be titles, or other noisy text. We generate two variants of each corpus: one that we tokenise with a BPE tokeniser, and another that retains UD-style tokenisation. The motivation behind this is to control for subwords: the absence of subword tokenisation is harder for our models to recover from, as they must be able to cluster tokens that have the same morphological affixes without explicit access to these affixes.

For our BPE segmented corpora, we use a model vocabulary of size 2048; this vocabulary is derived by training a fastBPE tokenizer on the respective training corpus. For UD-style tokenisation, we also use a vocab with 2048 unique tokens. We handle unknown tokens by replacing them with `<unk>` tokens; we also filter out sentences that have over 90% OOV tokens in the process of sentence selection, to avoid noise. As both our corpora are fairly noisy, we also apply a set of heuristics to eliminate corpus noise; for instance, we filter out sentences based on the number of title-cased tokens in them, to avoid scraping Wikipedia titles.

III.3.3 Perturbations

To adequately isolate the effects of word order and morphology, we apply three modifications to each combination of tokenisation method and corpus, giving us a total of $2 * 2 * 4 = 16$ corpora per language; with 15 languages and 5 seeds, this equates to $16 * 15 * 5 = 1200$ experiments in all.

Original Our original, unmodified corpus, presented with both UD- and BPE-based tokenisation.

Shuffled We modify our corpus by shuffling every sentence at a word level. Note that the shuffling procedure takes place before BPE segmentation, similar to Sinha et al. (2021). Ideally, given no word-order context, our masked language models should only be able to rely on morphological information, or bag-of-words distributions, in order to build a multilingual space. This also has a similar effect to removing positional embeddings from the transformer, as described in Sinha et al. (2021). Positional embeddings act as an ordering mechanism in masked language modelling; without them, a corpus is similar to our shuffled corpus.

Lemmatised We use the `LEMMA` Universal Dependencies field to generate our corpus, instead of the usual `FORM` field. The motivation here is to eliminate

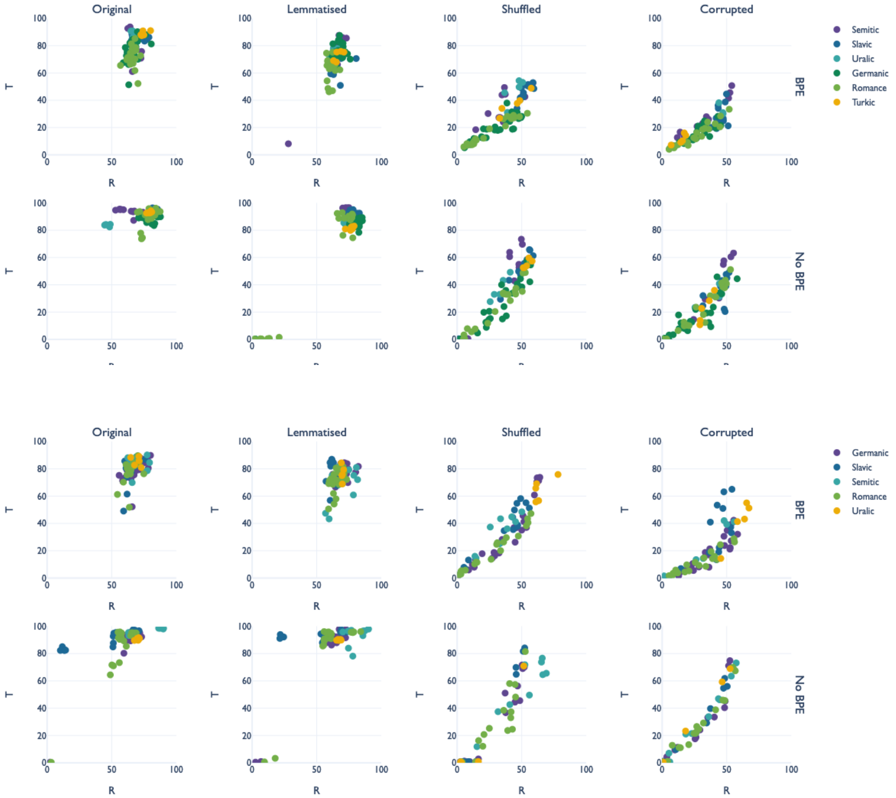


Figure III.1: Results for our four perturbations, with and without BPE, with data from Common Crawl (top) and Wikipedia (bottom). Scores (sentence retrieval on the X-axis, word translation on the Y-axis) are averaged over layers 0 and 8.

all morphological information; the difference between this and avoiding BPE tokenisation is that lemmatisation prevents unique word forms from having separate vocab indices.

Corrupted This corpus is both lemmatised and shuffled. Given this precondition, and UD-style tokenisation, there ought to be no information accessible to our model, beyond bag-of-word lemma statistics. We therefore expect word translation and sentence retrieval to be close to 0 in this setting.

III.3.4 Models and Evaluation

To evaluate our models' multilingual capabilities, we first train lower-capacity language models on each corpus. Each model is trained on the task of masked language modelling, on the concatenation of both halves (original and shifted) of a corpus. We use Dufter and Schütze (2021)'s BERT variant, which downsizes the original BERT model; we use single-headed, 12 layer transformer, with a head dimensionality of 64 and a feed-forward dimensionality of 256. This allows us to rapidly train a model on our corpora (in approximately 30–60 minutes per model). We set the random seed of each model to the same as the random seed used to generate the corpus we train it on; i.e. the model with seed 0, for English, is trained on the English corpus that was generated using a random seed of 0. Models are trained on V100 GPUs, each for approximately 1 hour.

Finally, we evaluate word translation and sentence retrieval scores for these models by using the deterministic gold labels, obtained by simply adding the vocab size (for translation) and by dividing the corpus into two halves and generating a sequential mapping (for retrieval). Note that this evaluation does not involve fine-tuning language models: we use the cosine similarity between either a word or a sentence and its fake parallel, for word translation and sentence retrieval resepectively. For word translation, we ensure that non-initial subwords are not included in the evaluation; while this is not ideal, none of our languages are morphologically prefixing, implying that the bulk of the semantic content is in the initial subword.

III.4 Results

We present results per language and experiment on Common Crawl (top) and Wikipedia (bottom) in Figure III.1. We begin by making a few general observations before moving on to study correlations with morphosyntactic and corpus factors.

'Fails' are frequent We note, first, that across most of our experiments, we have several 'fails', where our model effectively has near 0 retrieval and translation capacity. While this observation in isolation is somewhat meaningless – the model might have failed to learn effectively, either due to the random seed or due to the hyperparameters – the sheer number of experiments we run for each scenario makes these results more meaningful, when used as a comparison between training scenarios, as evidence that a certain scenario is likelier to result in a fail than another.

BPE makes word translation harder Despite controlling for non-initial subwords, using BPE tokenisation results in a drop in translation score for all our experiments. We hypothesise that this is due to common word-initial subwords being distributionally 'overloaded'; they are more likely to appear in a wider range of contexts than whole tokens are, due to the variety in consecutive subwords.

Multilingualism is robust to lemmatisation Perhaps somewhat unsurprisingly, lemmatisation does not significantly affect model scores, indicating that our model relies more on word order to build multilingual spaces. Interestingly,

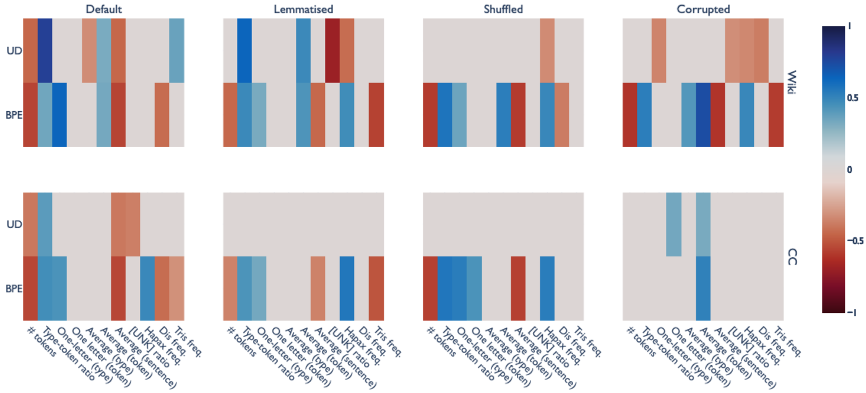


Figure III.2: Spearman correlations ($\alpha = 0.001$). Greyed-out values indicate insufficient evidence.

removing BPE segmentation results in an increase in fails on lemmatised corpora.

Bag-of-words is enough for (some) experiments Our most unexpected observation is that for both shuffling and corrupting, for both BPE and non-BPE, several experiments do appear to result in fairly successful retrieval/translation models, often with an accuracy higher than 50% on either task. This is surprising, given that a) this appears to contradict the findings of Dufter and Schütze (2021) about position embeddings being critical for multilingual spaces, and b) it implies that a simple bag-of-words model is enough to build a multilingual space. We attempt, in the following sections, to tease out what factors might enable this transfer. It is plausible that some part of this signal stems from the fact that the shuffling operation was carried out *prior* to BPE segmentation (Abdou et al., 2022); we discuss this further in Section III.5.4.

III.5 Analysis

III.5.1 Clustering

In order to find potential explanations for our results, we automatically cluster our scores, using retrieval and translation scores as our cluster metrics. To determine whether either languages (given that we have five experiments per language) or language families tend to actually represent logical, meaningful clusters, we set the number of clusters to be equivalent to the number of families, and use the adjusted Rand score (Vinh et al., 2010) to measure the distance between two clusterings – clusterings based on language/family, and learnt clusterings.

	Language		Family	
	BPE	UD	BPE	UD
Default	0.17/0.05	0.35/0.25	0.07/0.05	0.04/0.08
Lemmatised	0.16/0.11	**0.38**/0.14	0.10/0.04	**0.14**/0.07
Shuffled	0.15/0.13	0.03/0.01	0.07/0.10	0.02/0.05
Corrupted	0.14/0.12	0.05/0.02	0.13/0.09	0.01/0.02

Table III.2: Cluster similarities (adjusted Rand score) between language, or language family clusters, and k-means clustering, with a random seed of 42. Results on Wikipedia and Common Crawl are separated with a backslash.

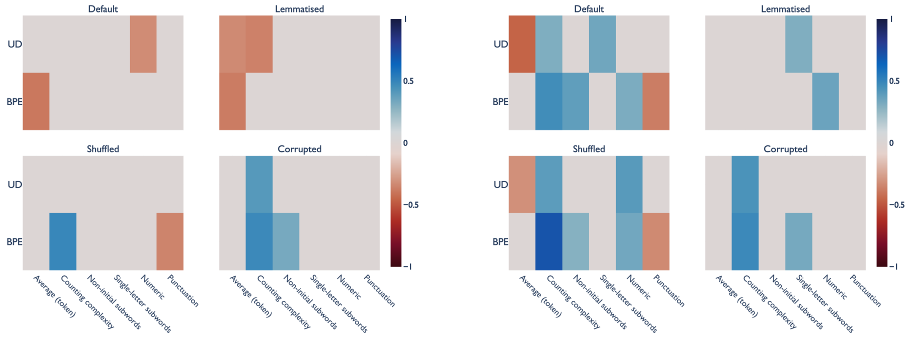
We present these results in Table III.2. First, clustering by language family shows little to no correlation with score-based clusters. Clusters of corpora in a single language ('language-based' clusters) are slightly clearer: while similarities are relatively low for all our BPE-based clusters, when we switch to UD tokenisation, the default and lemmatised cases begin to form more typologically relevant clusters, resembling languages. While these are by no means perfect overlaps, they are almost twice as realistic as for BPE-based tokenisation, implying that there exist language-specific features that correlate somewhat to the model's ability to form multilingual spaces. To investigate these findings in greater detail, we look for language-specific features – both corpus-specific features, and vocabulary features – and look for correlations that might explain our results.

III.5.2 Corpus correlations

We analyse our corpora, and measure correlations of model performance to a range of descriptive statistics, applied to the corpora that the models were trained on. For a single 'performance' metric, we follow Dufter and Schütze (2021) in defining a model's ML score as the average of its word translation and sentence retrieval scores, at layers 0 and 7. We measure correlations with:

- The number of training tokens
- The type-token ratio
- The number of one-letter types
- The number of one-letter tokens
- Average type length (in characters);
- Average token length
- Average sentence length
- Frequency of *hapax*, *dis* and *tris legomena*

We present these statistics in Figure III.2. A clear difference between doing nothing/lemmatising and shuffling/corrupting leaps out. With UD tokenisation, none of our corpus metrics correlates well with model performance, while BPE tokenisation consistently throws out a range of correlations. There is also a clear



(a) Sentence retrieval

(b) Word translation

Figure III.3: Spearman correlations, with a more relaxed $\alpha = 0.01$. X-axis indicates vocabulary statistics. Y-axis indicates tokenisation method. Correlations are on Common Crawl data, with the appropriate metric averaged at layers 0 and 7.

difference between Wikipedia and Common Crawl; in general, we find that correlations tend to be either weaker or less significant with Common Crawl than with Wikipedia. We hypothesise that this is due to Wikipedia being both more homogeneous and less noisy as a corpus.

Type-token ratio is a strong predictor For the default (and, to some extent, lemmatised) models, we find that type-token ratio has a strong positive correlation to ML-score (particularly retrieval), implying that lexical diversity enables better transfer. This is perhaps unsurprising – infrequent types might act as 'anchors', allowing easier transfer for their surrounding contexts. This is somewhat backed up by the disappearance of this metric in shuffled models.

Avg. token length predicts BPE performance Over our scrambled corpora, for both Wikipedia and Common Crawl,[1] it appears that average token length correlates strongly to downstream performance. The fact that this occurs for BPE tokenisation and not UD implies that this is likely a proxy for the number of BPE splits, rather than a realistic cross-linguistic measure; the more aggressive the BPE, the poorer the model. This is also somewhat backed up by the fact that the number of tokens inversely correlates to BPE performance; the shorter the average BPE split, the more the actual number of tokens in a corpus, for a given language.

Sentence length often correlates negatively This finding is consistent across all our BPE models;[1] longer sentence lengths (in tokens) imply poorer

[1] While exceptions to these observations exist, they disappear when we use a less restrictive $\alpha = 0.005$

multilingual scores. This is likely at least partially related to the previous observation – the longer the average token, the less aggressive the BPE, and the less aggressive the BPE, the shorter the average sentence.

Hapax/dis/tris ratios Results generally tend to correlate positively with the ratio of *hapax legomena* to the total number of tokens, when BPE tokenisation is used. This difference is likely due to the presence of more *morphemic* hapaxes in BPE-tokenised models: UD tokenisation is likely to result in a long tail of rarer morphological forms of rarer tokens. Curiously, this correlation, albeit weaker, is reversed for *dis* and *tris legomena*.

III.5.3 Vocabulary correlations

Next, we examine ML score correlations with different properties of the size 2048 UD/BPE vocabulary for each model. Note that as each model is trained with a unique corpus, each model has a unique vocabulary. Our features include:

- Average token length; for non-initial wordpieces, we do not include the length of the prefix.
- Counting complexity, using UniMorph (Kirov et al., 2020) to count the number of distinct morphological features in a given language.
- The frequency of single-letter vocab items.
- The frequency of digits in the vocab.
- The frequency of punctutation in the vocab.

We present these correlations in two heatmaps in Figures III.3a and III.3b. Some of our observations back up the observations in the previous section (eg. token length correlates inversely with ML score).

Counting complexity is complex Gratifyingly, the counting complexity metric (Sagot, 2013) appears to match Cotterell et al. (2018)'s observation, and is positively correlated with both retrieval and (to a larger extent) translation. Strangely, however, this correlation also appears to hold for both *corrupted* corpora; this is odd, as these corpora are lemmatised, implying the *absence* of inflectional morphology. It is plausible that this effect is still visible (albeit weakened) due to differences in the distribution of function words and stems, when compared with a language with *actual* differences in counting complexity; a language with strong case-marking, for instance, is likely to have a very different distribution of adpositions than a language without. This finding also backs up Mielke et al. (2019), who suggest that vocabulary-level measures may correlate better.

Specific tokens may act as anchors For the task of word translation, we notice that positive correlations tend to occur with the frequency of non-initial

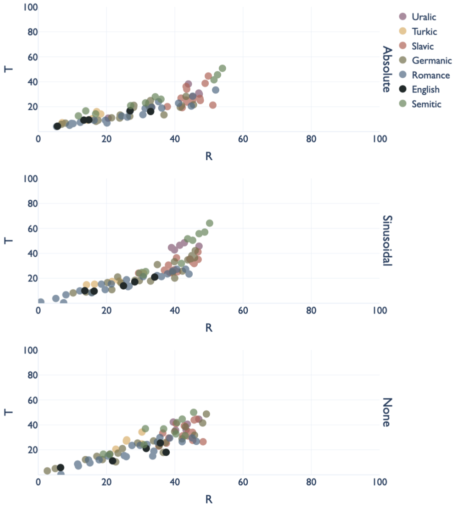


Figure III.4: Retrieval/translation scores for (learnt) absolute position, (fixed) sinusoidal position and no position. English in bold black for easier comparison with Dufter and Schütze (2021).

subwords, the frequency of digits, and the frequency of single-letter tokens. This effect, visible across all three categories, might indicate that these tokens act as anchors, enabling easier transfer in their contexts.

No clear patterns exist for retrieval We notice no clear factors contributing to retrieval. While the number of unused tokens does appear to correlate in the lemmatised models, this is mild and is likely to be an effect of the vocab size being effectively smaller.

III.5.4 Ablation experiments

While somewhat tangential to our original research question, we attempted to modify the positional embedding bias in our model. Dufter and Schütze (2021) show that positional embeddings are critical to building a multilingual space; Sinha et al. (2021) show that positional embeddings are critical to building *monolingual* language models, a finding backed up in other work (Abdou et al., 2022; Papadimitriou et al., 2022), where the authors also emphasise the importance of meaningful word order. These observations are somewhat contradictory to our findings, where shuffling corpora at a token-level still allows for successful multilingual space induction.

To resolve this, we train two additional models, on a corrupted variant of Common Crawl, presented in Figure III.4. The first of these has its learnt, absolute position embeddings (Devlin et al., 2019) replaced with sinusoidal embeddings, as in the original transformer paper (Vaswani et al., 2017), and the other has them removed entirely. While we would expect to see model performance drop considerably without position embeddings, this is often not the case at all; there is no real visible difference in performance across either of the tasks, implying that certain 'clues' are perhaps sufficient to build a multilingual space, even when a functional monolingual space might not exist for any of the languages.

Having said that, we note that English (annotated in black) is not one of the easier languages to build multilingual spaces for, even with absent position embeddings; as such, our English results are more similar to the results reported by Dufter and Schütze (2021).

III.6 Conclusion

In this work, we attempted to measure the variance in the ability of masked language models to build multilingual spaces with the underlying typology of the language. In doing so, we have shown that these models are capable of building multilingual spaces even when sentences are lemmatised and scrambled at a token level, showing that multilingualism can exist even when transformers act, functionally, like bag-of-words models. This does *not*, however, necessarily imply the ability to effectively model language (Abdou et al., 2022), but merely the ability to align two disjoint linguistic spaces.

We have also shown that, on the one hand, the ability to build a multilingual space is only weakly correlated to language (given multiple corpora) and to language family, and that, on the other hand, certain corpus-level metrics (specifically, type-token ratios and the presence of *hapax legomena*) are relatively good predictors of multilingual space quality, while others (such as the number of tokens or the average sentence length) are negatively correlated.

Our work is not without its caveats. For one, a lot of our correlating factors muddy the waters between what is an inherent property of the language itself, and what is a property of the *corpus* we use. While we use texts from the same domain in all our languages, both Wikipedia and Common Crawl are widely inconsistent across language, unless explicitly made comparable (Otero and López, 2010). Further, as discussed earlier, our scenario is not strictly realistic: first, this is a bilingual setup meant to approximate a multilingual one; second, both our languages have exactly the same structure; third, our language models are very underparameterised relative to full-scale models. It is unlikely that our observations would hold true in a real-world scenario; given, however, that our aim was to study the *inductive biases* of masked language models, using full-scale models would defeat the purpose somewhat, as the sheer volume of training data would have overridden these biases. Having said that, we present this work as an attempt to add to the often conflicting pool of papers attempting to shed some light on how language models acquire language.

Limitations

This work has several limitations, some of which we have addressed. To reiterate, in order to enable some degree of cross-linguistic diversity in this analysis, our bilingual setup is only an approximation of a true multilingual setup. Conversely, we are limited in the data we have access to: for inclusion in this study, languages had to have large and relatively noiseless dependency-parsed corpora available; as such, we are somewhat biased towards over-representing Indo-European languages.

Ethical considerations

The research presented in this work is compatible with the ACL ethics policy; the data we use is a toy subset of openly available corpora, and our models are very underparameterised, relative to the current state-of-the-art. Given the sheer number of models we train, our main experimental findings require approximately 1200 GPU hours for training, approximately equivalent to the amount of time required to train a full-scale BERT model on the same V100 GPUs.[2]

References

Abdou, M. et al. (2022). “Word order does matter and shuffled language models know it”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Artetxe, M. et al. (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.

Cotterell, R. et al. (2018). “Are All Languages Equally Hard to Language-Model?” In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics.

Deshpande, A. et al. (2021). “When Is BERT Multilingual? Isolating Crucial Ingredients for Cross-lingual Transfer”. In: *arXiv:2110.14782 [cs]*.

Devlin, J. et al. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805 [cs]*.

Dufter, P. and Schütze, H. (2021). “Identifying Necessary Elements for BERT’s Multilinguality”. In: *arXiv:2005.00396 [cs]*.

Gulordava, K. et al. (2018). “Colorless Green Recurrent Networks Dream Hierarchically”. In: *arXiv:1803.11138 [cs]*.

K, K. et al. (2020). “Cross-Lingual Ability of Multilingual BERT: An Empirical Study”. In: *arXiv:1912.07840 [cs]*.

[2] https://developer.nvidia.com/blog/training-bert-with-gpus/

Kirov, C. et al. (2020). “UniMorph 2.0: Universal Morphology”. In: *arXiv:1810.11101 [cs]*.

Levshina, N. (2019). “Token-based typology and word order entropy: A study based on Universal Dependencies”. In: *Linguistic Typology* vol. 23.

Linzen, T. et al. (2016). “Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies”. In: *Transactions of the Association for Computational Linguistics* vol. 4.

Marneffe, M.-C. de et al. (2021). “Universal Dependencies”. In: *Computational Linguistics* vol. 47, no. 2.

Mielke, S. J. et al. (2019). “What Kind of Language Is Hard to Language-Model?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Nivre, J. et al. (2016). “Universal dependencies v1: A multilingual treebank collection”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

Nivre, J. et al. (2020). “Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Otero, P. G. and López, I. G. (2010). “Wikipedia as multilingual source of comparable corpora”. In: *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora, LREC*. Citeseer.

Papadimitriou, I. et al. (2022). “When Classifying Arguments, BERT Doesn't Care About Word Order... Except When It Matters”. In: *Proceedings of the Society for Computation in Linguistics* vol. 5, no. 1.

Pires, T. et al. (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.

Ravfogel, S. et al. (2019). “Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages”. In.

Sagot, B. (2013). “Comparing complexity measures”. In: *Computational approaches to morphological complexity*.

Sinha, K. et al. (2021). “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Vaswani, A. et al. (2017). “Attention Is All You Need”. In: *arXiv:1706.03762 [cs]*.

Vinh, N. X. et al. (2010). “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance”. In: *Journal of Machine Learning Research* vol. 11, no. 95.

White, J. C. and Cotterell, R. (2021). “Examining the Inductive Bias of Neural Language Models with Artificial Languages”. In: *arXiv:2106.01044 [cs]*.

Paper IV

Probing Multilingual Sentence Representations with X-Probe

Vinit Ravishankar, Lilja Øvrelid, Erik Velldal

Appears in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, August 2019, pp. 156-168. DOI: 10.18653/v1/W19-4318.

Abstract

This paper extends the task of probing sentence representations for linguistic insight in a multilingual domain. In doing so, we make two contributions: first, we provide datasets for multilingual probing, derived from Wikipedia, in five languages, viz. English, French, German, Spanish and Russian. Second, we evaluate six sentence encoders for each language, each trained by mapping sentence representations to English sentence representations, using sentences in a parallel corpus. We discover that cross-lingually mapped representations are often better at retaining certain linguistic information than representations derived from English encoders trained on natural language inference (NLI) as a downstream task.

Contents

The authors were partially supported by CERN.

IV.1 Introduction

In recent years, there has been a considerable amount of research into attempting to represent contexts longer than single words with fixed-length vectors. These representations typically tend to focus on attempting to represent sentences, although phrase- and paragraph-centric mechanisms do exist. These have moved well beyond relatively naïve compositional methods, such as additive and multiplicative methods (Mitchell and Lapata, 2008), one of the earlier papers on the subject. There have been several proposed approaches to learning these representations since, both unsupervised and supervised. Naturally, this has also sparked interest in evaluation methods for sentence representations; the focus of this paper is on *probing*-centric evaluations, and their extension to a multilingual domain.

In Section V.2, we provide a literature review of prior work in the numerous domains that our paper builds upon. Section IV.3 motivates the principle of cross-lingual probing and describes our goals. In Section VI.4, we describe our probing tasks and relevant modifications, if any. Section IV.5 describes our sentence encoders, as well as the procedure we follow for training, mapping and probing. Section IV.6 describes our data and relevant preprocessing methods we applied. Section IV.7 presents a detailed evaluation from several perspectives, which we discuss in Section IV.8. We conclude, as well as describe avenues for future work, in Section IV.9. Our hyperparameters are described in Appendix IV.A, and further detailed results that are not critical to the paper are tabulated in IV.B.

IV.2 Background

IV.2.1 Sentence representation learning

Numerous methods for learning sentence representations exist. Many of these methods are unsupervised, and thus do not have much significant annotation burden. Most of these methods are, however, structured: they rely on the sentences in training data being ordered and not randomly sampled. The aptly named *SkipThoughts* (Kiros et al., 2015) is a well-known earlier work, and uses recurrent encoder-decoder models to 'decode' sentences surrounding the encoded sentence, using the final encoder state as the encoded sentence's representation. Cer et al. (2018) evaluate two different encoders, a deep averaging network and a transformer, on unsupervised data drawn from a variety of web sources. Hill et al. (2016) describe a model based on denoising auto-encoders, and a simplified variant of SkipThoughts, that sums up source word embeddings, that they dub (*FastSent*). Another SkipThoughts variant (Logeswaran and Lee, 2018) uses a multiple-choice objective for contextual sentences, over the more complicated decoder-based objective.

Several supervised approaches to building representations also exist. An earlier work is Charagram (Wieting et al., 2016), which uses paraphrase data and builds on character representations to arrive at sentence representations. More recent papers use a diverse variety of target tasks to ground representations, such as visual data (Kiela et al., 2018), machine translation data (McCann et al., 2017), and even multiple tasks, in a multi-task learning framework (Subramanian et al., 2018). Relevant to this paper is Conneau et al.'s (2017) *InferSent*, that uses natural language inference (NLI) data to ground representations: they learn these representations on the well-known SNLI dataset (Bowman et al., 2015).

IV.2.2 Multilingual representations

Whilst sentence representation is a thriving research domain, there has been relatively less work on multilingualism in the context of sentence representation learning: most prior work has been focussed on multilingual word representation. For sentence representations, an early work (Schwenk and Douze, 2017) proposes a seq2seq-based objective, using machine learning encoders to map source sequences to fixed-length vectors. Along similar lines, **conneau_xnli:_2018** propose using machine translation data to transfer sentence representations pre-trained on NLI, using a mean squared error (MSE) loss - this is the approach we follow.

Artetxe and Schwenk (2019) present a 'language agnostic' sentence representation system learnt over machine translation; the agnosticism refers to the joint BPE vocabulary that they construct over all languages, giving their encoders no language information, whilst their decoders are told what language to generate. Similarly, Lample and Conneau (2019) present pretrained cross-lingual models (XLM), based on modern pretraining mechanisms; specifically, a variant of the masked LM pretraining scheme used in BERT (Devlin et al., 2019).

Contemporaneous with this work, Aldarmaki and Diab (2019) present an evaluation of three cross-lingual sentence transfer methods. Their methods include joint cross-lingual modelling methods that extend monolingual objectives to cross-lingual training, representation transfer learning methods that attempt to 'optimise' sentence representations to be similar to parallel representations in another language, and sentence mapping methods based on orthogonal word embedding transfer: the authors use a parallel corpus as a 'seed dictionary' to fit a transformation matrix between their source and target languages.

IV.2.3 On evaluation

Work on evaluating sentence representations was encouraged by the release of the SentEval toolkit (Conneau and Kiela, 2018), which provided an easy-to-use framework that sentence representations could be 'plugged' into, for rapid downstream evaluation on numerous tasks: these include several classification tasks, textual entailment and similarity tasks, a paraphrase detection task, and caption/image retrieval tasks. Conneau et al. (2018) also created a set of 'probing tasks', a variant on the theme of diagnostic classification (Belinkov et al., 2017; Hupkes et al., 2018), that would attempt to quantify precisely what sort of linguistic information was being retained by sentence representations.

The authors, whose work focussed on evaluating representations for English, provided Spearman correlations between the performance of a particular representation mechanism on being probed for specific linguistic properties, and the downstream performance on a variety of NLP tasks. Along similar lines, and contemporaneously with this work, Liu et al. (2019) probe three pretrained contextualised word representation models – ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and the OpenAI transformer (Radford et al., 2018) – with a "suite of sixteen diverse probing tasks".

On a different note, Saphra and Lopez (2019) present a CCA-based method to compare representation learning dynamics across time and models, without explicitly requiring annotated probing corpora. They motivate the use of SVCCA (Raghu et al., 2017) to quantify precisely what an encoder learns by comparing the representations it generates with representations generated by an architecture trained specifically for a certain task, with the intuition that a higher similarity between the representations generated by the generic encoder and the specialised representations would indicate that the encoder is capable of encapsulating more task-relevant information. Their method has numerous advantages over traditional diagnostic classification, such as the elimination of the classifier, which reduces the risk of an additional component obfuscating results.

A visible limitation of the datasets provided by these probing tasks is that most of them were created with the idea of evaluating representations built for English language data. In this spirit, what we propose is analogous to Abdou et al.'s (2018) work on generating multilingual evaluation corpora for word representations. Within the realm of evaluating *multilingual* sentence representations, **conneau_xnli:_2018** describe the XNLI dataset, a set of translations of the development and test portions of the multi-genre MultiNLI inference dataset (Williams et al., 2018). This, in a sense, is an extension of a predominantly monolingual task to the multilingual domain; the authors evaluate sentence representations derived by *mapping* non-English representations to an English representation space.

The original XNLI paper provides a baseline representation mapping technique, based on minimising the mean-squared error (MSE) loss between sentence representations across a parallel corpus. Their English language sentence representations are derived from an encoder trained on NLI data (Conneau et al., 2017), and their target language representations are randomly initialised for a parallel sentence. While this system does perform reasonably well, a more naive machine-translation based approach performs better.

IV.3 Multilingual evaluation

The focus of this paper is twofold. First, we provide five datasets for probing mapped sentence representations, in five languages (including an additional dataset for English), drawn from a different domain to Conneau et al.'s probing dataset: specifically, from Wikipedia. Second, we probe a selection of mapped sentence representations, in an attempt to answer precisely what linguistic features are retained, and to what extent, post mapping. The emphasis of this evaluation is therefore, crucially, not a probing-oriented analysis of representations *trained* on different languages, but an analysis of the effects of MSE-based mapping procedures on the ability of sentence representations to retain linguistic features. In this sense, our focus is less on the correlation between probing performance and downstream performance, and more on the relative performance of our representations on probing tasks.

Despite having described (in Section V.2) numerous methods, both for learning monolingual sentence representations, and for mapping them cross-linguistically, we restrict our work to a smaller subset of these. Specifically, we evaluate six encoders, each trained in a supervised fashion on NLI data.

Whilst our choice of languages could have been more typologically diverse, we were restricted by three factors:

1. the availability of a parallel corpus with English for our mapping procedure
2. the availability of a large enough Wikipedia to allow us to extract sufficient data (for instance, the Arabic Wikipedia was not large enough to fully extract data for all our tasks)
3. the inclusion of the language in XNLI. Despite not being necessary, we believed it would be interesting to have a 'real' downstream task to compare to.

IV.4 Probing

We use most of the probing tasks described in Conneau et al. (2018). Due to the differences in corpus domain, we alter some of their word-frequency parameters. We also exclude the top constituent (**TopConst**) task; we noticed that Wikipedia tended to have far less diversity in sentence structure than the original Toronto Books corpus, due to the more encyclopaedic style of writing. A brief description of the tasks follows, although we urge the reader to refer to the original paper for more detailed descriptions.

1. Sentence length: In **SentLen**, sentences are divided into multiple bins based on their length; the job of the classifier is to predict the appropriate bin, creating a 6-way classification task.
2. Word count: In **WC**, we sample sentences that feature exactly one amongst a thousand mid-frequency words, and train the classifier to predict the word: this is the most 'difficult' task, in that it has the most possible classes.
3. Tree depth: The **TreeDepth** task simply asks the representation to predict the depth of the sentence's syntax tree. Unlike the original paper, we use

the depth the of the dependency tree instead of the constituency tree: this has the added benefit of being faster to extract, due to the relative speed of dependency parsing, as well as having better multilingual support. We also replace the authors' sentence-length-decorrelation procedure with a naïver one, where we sample an equal number of d-depth trees for each sentence length bin.

4. Bigram shift: In **BiShift**, for half the sentences in the dataset, the order of words in a randomly sampled bigram is reversed. The classifier learns to predict whether or not the sentence contains a reversal.

5. Subject number: The **SubjNum** task asks the classifier to predict the number of the subject of the head verb of the sentence. Only sentences with exactly one subject (annotated with the `nsubj` relation) attached to the root verb were considered.

6. Object number: **ObjNum**, similar to the subject number task, was annotated with the number of the direct object of the head verb (annotated with the `obj` relation).

7. Coordination inversion: In **CoordInv**, two main clauses joined by a coordinating conjunction (annotated with the `cc` and `conj` relations) have their orders reversed, with a probability of one in two. Only sentences with exactly two top-level conjuncts are considered.

8. (Semantic) odd man out: **SOMO**, one of the more difficult tasks in the collection, replaces a randomly sampled word with another word with comparable corpus bigram frequencies, for both bigrams formed with the preceding and the succeeding words. We defined 'comparable' as having a log-frequency difference not greater than 1.

9. Tense prediction: The **Tense** prediction asks the classifier to predict the tense of the main verb: the task uses a rather simple division of tenses; two tenses, `Past` and `Pres`. Tense information was extracted from UD morphological annotation.

IV.5 Encoders

The NLI-oriented training approach for all our encoders is based on *InferSent* (Conneau et al., 2017). Our first encoder is an RNN-based encoder (specifically, an LSTM); we use two variants of this encoder, one that uses max-pooling over bidirectional RNN states, and another that uses the last recurrent state. Our second encoder is a self-attention based encoder Lin et al. (2017), with the same max-pool/last-state variants. Finally, we include a convolutional sentence representation model inspired by Gan et al. (2017); this model has an order of magnitude fewer parameters than the RNN- and attention-based variants. A variant of this CNN-based encoder has the maximum pooling replaced with average pooling.

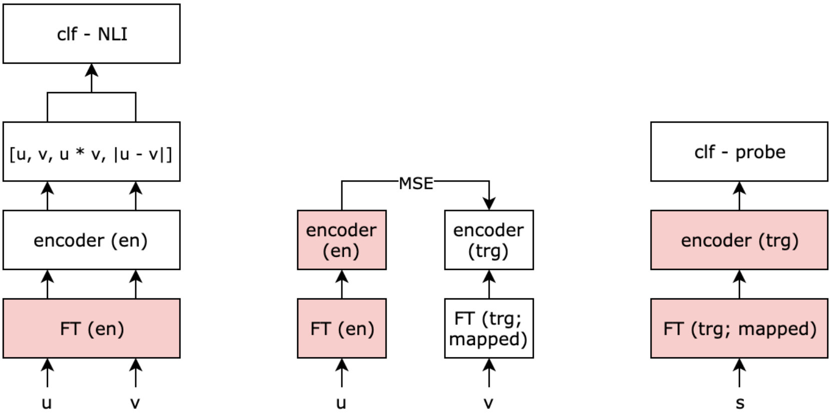


Figure IV.1: (a) an English-language encoder is trained on NLI data; (b) parallel sentences are encoded in English and the target language, and the MSE loss between them is minimised; (c) the mapped target encoders are used downstream in probing. Greyed-out blocks represent 'frozen' components that do not further adjust their parameters.

IV.5.1 Representation learning

We train all our encoders to represent sentences using the same NLI-based objective followed by Conneau et al. (2017). More precisely, we first convert the word indices for both our premise and our hypothesis into dense word representations using pretrained fastText word embeddings (Bojanowski et al., 2017). These representations are then passed to our encoder of choice, which returns two fixed-length vectors: u for the premise, and v for the hypothesis. These vectors are combined and concatenated, as $[u, v, u * v, \mid u - v \mid]$, and then passed through a classifier with a softmax layer that outputs a probability distribution over the three NLI labels.

IV.5.2 Mapping

Our procedure for mapping our encoders cross-linguistically broadly follows the principled mapping approach described in **conneau_xnli:_2018**. The procedure begins by mapping our *word* representations to the same vector space. Unlike the original paper, we use the supervised variant of VecMap (Artetxe et al., 2016) for representation mapping; however, we use seed dictionaries described in Lample et al. (2018). Having mapped our word representations, we proceed to map our sentence representations. To do so, we first build an English-language encoder, using (frozen) word representations and (frozen) encoder weights obtained in Section IV.5.1. We then build a target language encoder, using word embeddings mapped to the same space as the English embeddings. The sentence encoder itself is initialised with random parameters.

We then encode the source and target sentences in an en-trg machine translation corpus, where trg is our target language. Our English encoder returns a 'meaningful' representation: recall that the encoder has weights trained on NLI data. We then use a mean-square error loss function to reduce the distance between our target-language representation and the English representation; the system then backpropagates through the target language encoder to obtain better parameters.

Our MSE loss function, similar to **conneau_xnli:_2018**'s function, attempts to minimise the distance between representations of parallel sentences, whilst simultaneously maximising the distance between random sentences sampled from either language in the pair. Mathematically, the alignment loss is given by:

$$\mathcal{L}_{align} = ||\mathbf{x} - \mathbf{y}||_2 - \lambda(||\mathbf{x_c} - \mathbf{y}||_2 + ||\mathbf{x} - \mathbf{y_c}||_2)$$

where λ is a hyperparameter.

We evaluate our mapped encoder on the relevant validation data section from the XNLI corpus per epoch, and terminate the mapping procedure when our validation accuracy does not improve for two consecutive epochs.

IV.5.3 Multilingual probing

Having obtained our mapped sentence representation encoder, we proceed to plug the encoder into our probing architecture downstream, and evaluate classifier performance.

First, we load our mapped word representations for the language that we intend to analyse. We use these word representations to build sentence representations, using the encoder architecture of choice. We then add a simple multi-layer perceptron (MLP) that learns to predict the appropriate label for each task: the MLP consists of a dense layer, a non-linearity (we use the sigmoid function), and another dense layer that we softmax over to arrive at per-class probabilities. During training, we keep the encoder's parameters fixed. Mathematically, therefore, given an encoder f with parameters θ, and word representations $\boldsymbol{w_k}$ for each word k:

$$\begin{aligned}\boldsymbol{s} &= f(\boldsymbol{w_0}, \boldsymbol{w_1}, ..., \boldsymbol{w_n}; \theta)\\ \boldsymbol{z} &= \mathrm{MLP}(s)\\ \boldsymbol{y} &= \mathrm{softmax}(\boldsymbol{z})\end{aligned}$$

where 'MLP' refers to a multi-layer perceptron with one sigmoid hidden layer.

Finally, we evaluate our representations on the relevant test portion. Whilst Conneau et al. used grid search to determine the best hyperparameters for each probing task, we did not do so, due to both time constraints, and in an attempt to ensure classifier uniformity across languages. We describe our probing results in Section IV.7.

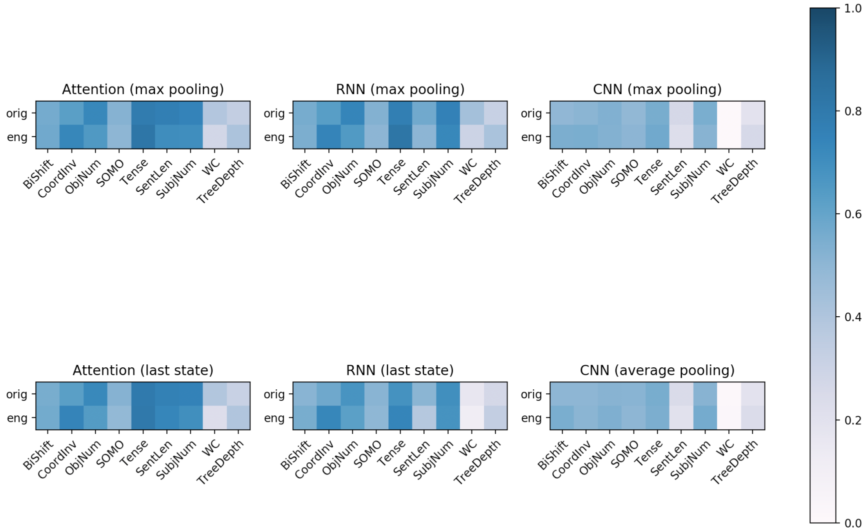


Figure IV.2: Probing accuracies for our six encoders on Conneau et al.'s dataset (`orig`), compared to our Wikipedia-derived dataset (`eng`)

IV.6 Data

IV.6.1 Probing data

We build our probing datasets using the relevant language's Wikipedia dump as a corpus. Specifically, we use Wikipedia dumps (dated 2019-02-01), which we process using the WikiExtractor utility[1]. We use the Punkt tokeniser (Kiss and Strunk, 2006) to segment our Wikipedia dumps into discrete sentences. For Russian, which lacked a Punkt tokenisation model, we used the UDPipe (Straka and Straková, 2017) toolkit to perform segmentation.

Having segmented our data, we used the Moses (Koehn et al., 2007) tokeniser for the appropriate language, falling back to English tokenisation when unavailable: this was similar to XNLI's tokenisation schema, and therefore necessary for appropriate evaluation on XNLI.

Next, we obtained dependency parses for our sentences, again using the UDPipe toolkit's pretrained models, trained on Universal Dependencies treebanks (Nivre et al., 2015). We then processed these dependency parsed corpora to extract the appropriate sentences; each task had 120k extracted sentences, divided into training/validation/test splits with 100k, 10k and 10k sentences respectively.

[1] https://github.com/attardi/wikiextractor/

IV.6.2 Mapping data

For mapping our sentence representations, we were restricted by the availability of large parallel corpora we could use for our mapping procedure. We used two such corpora: the Europarl corpus (Koehn, 2005), a multilingual collection of European Parliament proceedings, and the MultiUN corpus (Tiedemann, 2012), a collection of translated documents from the United Nations. We used Europarl for the official EU languages we analysed: German and Spanish. For Russian, we used MultiUN. We used both corpora for French, to attempt to analyse what, if any, effect the mapping corpus would have. We also truncated our MultiUN cororpora to 2 million sentences, to keep the corpus size roughly equivalent to Europarl, and also due to time and resource constraints: mapping representations on the complete 10 million sentence corpus would have required significant amounts of time.

Both our corpora were pre-segmented: we followed the same Moses-based tokenisation scheme that we did for our probing corpora, falling back to English for languages that lacked appropriate tokeniser models.

IV.7 Evaluation

As a preface to this section, we reiterate that the goal of this work was not to attempt to reach state-of-the-art on the tasks we describe; our goal was primarily to study the effect of transfer on sentence representations.

Our first step during evaluation, therefore, was to probe all our encoders using Conneau et al.'s original probing corpus, and compare these results to our English-language results on our Wikipedia-generated corpus. We present these results in the form of a heatmap in Figure IV.2.

Similarities between results on our corpora are instantly visible; these also appear to hold across encoders. Tasks with minor visible differences include **WC**, the most 'difficult' classification task (1k classes), and **TreeDepth**, where we use dependency tree depth instead of constituency tree depth, as well as a different sampling mechanism.

Next, we present Spearman correlations between the performance of our encoders on probing tasks and on the only 'true' cross-lingual downstream task we evaluated our systems on: cross-lingual natural language inference, via the XNLI (**conneau_xnli:_2018**) corpus. A caveat here is that we make no claims about the statistical significance of these results; given just six data points per language per task, our p-values tend to be well below acceptable for statistical significance. We refer the reader to Conneau et al.'s original probing work, where despite having results for 30 encoders, correlations between many downstream and probing tasks were not statistically significant. Our correlations are presented, again in the form of a heatmap, in Figure IV.3. Our absolute results on XNLI are presented in the appendix. These are not a focus for this work: we did not attempt to obtain state-of-the-art, nor, indeed, perform any sort of hyperparameter optimisation to get the 'best' possible results. Given these caveats, we draw the reader's attention to the fact that the overwhelming majority of correlations are negative.

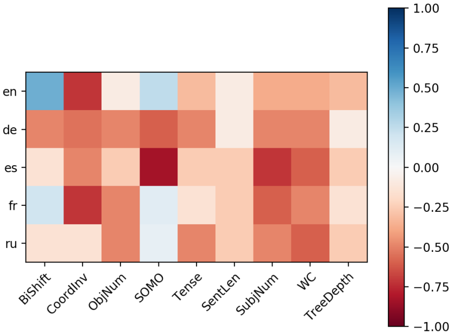


Figure IV.3: Spearman correlation between probing performance and XNLI; results are not statistically significant.

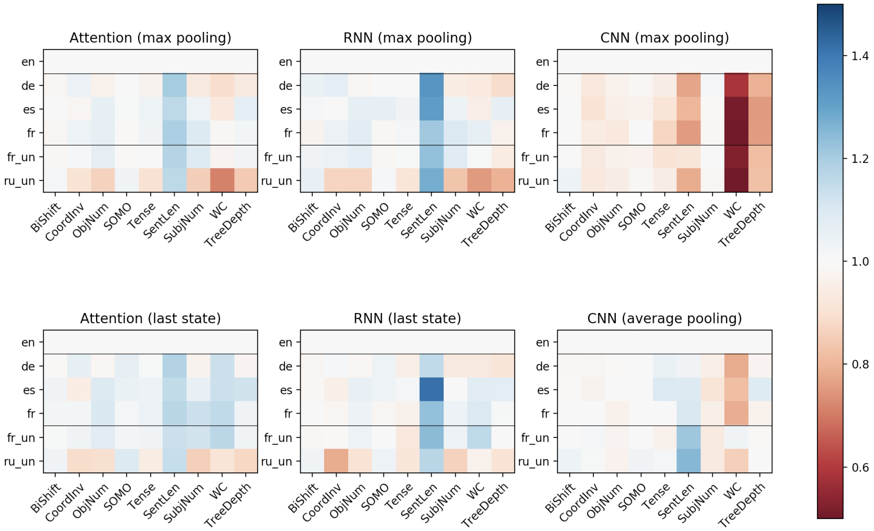


Figure IV.4: Probing results for each encoder relative to results on English. The second horizontal line indicates a switch in corpora. A white square indicates a value of 1, i.e. a parity in performance

Finally, and most importantly, we measure downstream performance on probing tasks for all our cross-lingually mapped encoders. For visualisation relevant to our goals, and for brevity, we present these results, in Figure IV.4, as a heatmap of probing results *relative* to (our) English probing results; a full table with numeric results is presented in Appendix IV.B.

IV.8 Discussion

Our cross-lingual results display some very interesting characteristics, that we enumerate and attempt to explain in this section. These results can be analysed along three dimensions: that of language, encoder mechanism, and the probing task itself.

IV.8.1 Language

Whilst our results are broadly similar across languages, Russian appears to be an exception to this: our probing performance for most tasks is considerably worse when transferred to Russian than other languages. Transfer corpus does not appear to be a factor in this case: most of our encoders perform very similarly on both the Europarl and the UN variants of our transferred French representations. These are interesting preliminary results, that would require further analysis: as we mentioned in an earlier section, we were rather limited in our choice of languages, however, we foresee a possible extension to this work including more typologically diverse languages. One possible explanation for the relatively poor results on Russian is the nature of the word embeddings themselves: whilst we did not use the same methods, we did map our embeddings to the same space using the same dictionaries as Lample et al. (2018). The results they describe for word translation retrieval are considerably poorer for English and Russian than they are for English and Spanish, French or German.

IV.8.2 Probing task

An immediate surprising takeaway from our results is the (perhaps counter-intuitive) fact that transferred representations are not necessarily worse at probing tasks than trained representations are. To help with the analysis of Figure IV.4, we present Table IV.1, where several trends are easily visible. In particular, a task that appears to stand out is **SentLen**, with transferred encoders displaying considerably improved performance in five out of six cases.

Apart from sentence length, both number prediction tasks – **SubjNum** and **ObjNum** – show noticeable improvements when transferred to a non-English language. The fact that this improvement is consistent across both number tasks likely also rules out mere coincidence. We hypothesise that the explanation for these three tasks in particular showing improvements on transfer lies in the specific nature of the mapping task. While it is plausible that this is due to these specific phenomena being less critical to NLI (on which our English encoders were trained) than to the attempt made by our target encoders to *emulate* these English representations, it is not immediately clear how these encoders are capable of exceeding the predictive capabilities of the encoders they are attempting to emulate.

Another interesting observation is the variance in performance for the word content (**WC**) task, which also happens to be the 'hardest' task with the most output classes. We further note that, regrettably, none of our encoders were able to learn anything on **SOMO**.

Task	μ	σ
BiShift	0.558	0.013
CoordInv	0.656	0.111
ObjNum	0.605	0.073
SOMO	0.505	0.011
Tense	0.708	0.124
SentLen	0.523	0.259
SubjNum	0.643	0.099
WC	0.152	0.115
TreeDepth	0.330	0.082

Table IV.1: Mean and standard deviations for the absolute performance for each probing task, across languages and encoders

IV.8.3 Encoder

All our encoders do appear to display very distinctive probing patterns, with variants of each encoder being more similar to each other than to different encoders. We enumerate some of the key observations:

1. Both our CNNs appear to perform worse than attentive or recurrent mechanisms; this is, however, perfectly understandable, as our CNN-based models had an order of magnitude fewer parameters than the recurrent ones. The choice of pooling mechanism, however, appears to have a more significant effect on convolutional encoders than on others.

2. Attentive encoders appear to be better at probing in general, whilst recurrent encoders show extremely strong performance on certain tasks, such as sentence length.

3. The max-pooled CNN is the only encoder that shows considerably worse performance on sentence length. This is also true for English, as is visible from Figure IV.2. We hypothesise that the fixed-length filters used in convolutional encoders do not store much information about sentence length, as they only observe chunks of the sentence. A max-pooling mechanism further compounds this inability to store length by eliminating possible compositional length information that mean-pooling does ignore.

IV.9 Conclusions

Our analysis reveals several interesting patterns that appear to hold during cross-lingual transfer. Several of our probing tasks give us clearer insight into the sentence representations that we have generated by cross-lingual mapping, which is much needed: the principle of learning a sentence representation in parallel, combined with the fact that these representations actually appear to 'work' downstream, raises a lot of questions both about what information sentence representations hold, but more interestingly, in a cross-lingual context, about what *mutual* information a sentence and its translation contain.

We open-source both our training code and the probing datasets (that we dub X-PROBE)[2] that we generated in the hope that the domain of cross-lingual analysis sees further work. There are several avenues for expansion, the most obvious being a probing-oriented analysis of more complex contextualisers, such as BERT, as well as of massively multilingual or language agnostic model.

We also hypothesise that more can be said about probing with a different selection of probing tasks; indeed, Liu et al. (2019) do provide a set of tasks that do not overlap with the tasks we have used. Selecting probing tasks that might tell allow us to better interpret cross-lingual modelling is another logical path one might follow. On a similar theme, an interesting research direction also involve adaptations of simple probing tasks describing linguistic phenomena to specialised architectures, for better comparison using SVCCA-style analyses (Saphra and Lopez, 2019).

Finally, we would also like to expand these datasets to more typologically diverse languages. A challenge in doing so is the availability of corpora that are large enough; none of our probing tasks have any sentences in common, which, given the size of each task's corpus, requires a fairly large corpus for extraction. However, this process could possibly be simplified massively by removing this mutual exclusivity requirement, which would vastly simplify the process.

Acknowledgements. The first author was partially supported by The Research Council of Norway.

References

Abdou, M. et al. (2018). "MGAD: Multilingual Generation of Analogy Datasets". In: *Proceedings of the 11th Language Resources and Evaluation Conference*. Miyazaki, Japan: European Language Resource Association.

Aldarmaki, H. and Diab, M. (2019). "Scalable Cross-Lingual Transfer of Neural Sentence Embeddings". In: *arXiv:1904.05542 [cs]*. arXiv: 1904.05542.

Artetxe, M. and Schwenk, H. (2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computational Linguistics* vol. 7.

[2] https://github.com/ltgoslo/xprobe

Artetxe, M. et al. (2016). “Learning principled bilingual mappings of word embeddings while preserving monolingual invariance”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Belinkov, Y. et al. (2017). “What do Neural Machine Translation Models Learn about Morphology?” en. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics.

Bojanowski, P. et al. (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* vol. 5.

Bowman, S. R. et al. (2015). “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics.

Cer, D. et al. (2018). “Universal Sentence Encoder for English”. In.

Conneau, A. and Kiela, D. (2018). “SentEval: An Evaluation Toolkit for Universal Sentence Representations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Conneau, A. et al. (2017). “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics.

Conneau, A. et al. (2018). “What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties”. en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Devlin, J. et al. (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.

Gan, Z. et al. (2017). “Learning Generic Sentence Representations Using Convolutional Neural Networks”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics.

Hill, F. et al. (2016). “Learning Distributed Representations of Sentences from Unlabelled Data”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics.

Hupkes, D. et al. (2018). “Visualisation and ’diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure”. In: *Journal of Artificial Intelligence Research* vol. 61.

Kiela, D. et al. (2018). “Learning Visually Grounded Sentence 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.

Kiros, R. et al. (2015). “Skip-Thought Vectors”. In: *Advances in Neural Information Processing Systems* vol. 28.

Kiss, T. and Strunk, J. (2006). “Unsupervised multilingual sentence boundary detection”. In: *Computational Linguistics* vol. 32, no. 4.

Koehn, P. (2005). “Europarl: A parallel corpus for statistical machine translation”. In: *MT summit*. Vol. 5.

Koehn, P. et al. (2007). “Moses: Open source toolkit for statistical machine translation”. In: *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*.

Lample, G. and Conneau, A. (2019). “Cross-Lingual Language Model Pretraining”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lample, G. et al. (2018). “Word Translation Without Parallel Data”. In: *International Conference on Learning Representations*.

Lin, Z. et al. (2017). “A structured self-attentive sentence embedding”. In: *International Conference on Learning Representations*. International Conference on Learning Representations, ICLR.

Liu, N. F. et al. (2019). “Linguistic Knowledge and Transferability of Contextual Representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Logeswaran, L. and Lee, H. (2018). “An efficient framework for learning sentence representations”. In: *International Conference on Learning Representations*.

McCann, B. et al. (2017). “Learned in Translation: Contextualized Word Vectors”. In: NIPS’17.

Mitchell, J. and Lapata, M. (2008). “Vector-based Models of Semantic Composition”. In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics.

Nivre, J. et al. (2015). *Universal Dependencies 1.2*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Peters, M. et al. (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)*.

Radford, A. et al. (2018). *Improving language understanding by generative pre-training*.

Raghu, M. et al. (2017). “SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability”. In: *Advances in Neural Information Processing Systems*.

Saphra, N. and Lopez, A. (2019). “Understanding Learning Dynamics Of Language Models with SVCCA”. 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.

Schwenk, H. and Douze, M. (2017). "Learning Joint Multilingual Sentence Representations with Neural Machine Translation". In: *ACL 2017*.

Straka, M. and Straková, J. (2017). "Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe". In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver, Canada: Association for Computational Linguistics.

Subramanian, S. et al. (2018). "Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning". In: *International Conference on Learning Representations*.

Tiedemann, J. (2012). "Parallel Data, Tools and Interfaces in OPUS". In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Chair), N. C. (et al. Istanbul, Turkey: European Language Resources Association (ELRA).

Wieting, J. et al. (2016). "Charagram: Embedding Words and Sentences via Character n-grams". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics.

Williams, A. et al. (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". en. 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.

Appendix IV.A Hyperparameters

Component	**Layer**	**Value**
Global	Embeddings	300 (FastText)
	Batch size	10
	Optimiser	Adam
	Learning rate	10^{-3}
RNN	biLSTM dim	512
	biLSTM layers	2
	Dropout	10%
CNN	Filter sizes	(3, 4, 5)
	Padding	(1, 2, 2)
	Channels	800
	Projection dim	1024
Attention	biLSTM dim	512
	biLSTM layers	2
	Dropout	10%
	MLP dim	150
	Activation	tanh
	Attn. heads	60
Mapper	λ	0.25
Probe classifier	Hidden dim	150
	Activation	σ

Table IV.2: Hyperparameters, divided by the 'component' that each layer belongs to. Note that biRNN dims are per direction.

Appendix IV.B Additional results

Encoder	**Language**					
	English	German	Spanish	French	French (UN)	Russian
RNN (maxpool)	**0.71**	**0.66**	0.68	0.68	0.65	0.61
RNN (last)	0.66	0.63	0.65	0.65	0.63	0.59
CNN (maxpool)	0.51	0.39	0.41	0.36	0.44	0.43
CNN (avg. pool)	0.51	0.50	0.51	0.50	0.50	0.48
Attn. (maxpool)	**0.71**	0.64	0.67	0.67	0.67	0.60
Attn. (last)	0.70	0.65	**0.69**	**0.69**	**0.66**	**0.62**

Table IV.3: Language-specific results on relevant XNLI splits for each encoder

English	BiShift	CoordInv	ObjNum	SOMO	Tense	SentLen	SubjNum	WC	TreeDepth
Attention (maxpool)	0.57	0.73	0.65	0.5	0.82	0.7	0.7	0.27	0.41
Attention (last)	0.56	0.74	0.64	0.49	0.8	0.74	0.7	0.22	0.4
RNN (maxpool)	0.54	0.74	0.65	0.5	0.82	0.51	0.73	0.3	0.42
RNN (last)	0.55	0.73	0.62	0.5	0.74	0.38	0.68	0.11	0.34
CNN (maxpool)	0.55	0.55	0.53	0.51	0.57	0.22	0.52	0.01	0.26
CNN (avg. pool)	0.55	0.51	0.54	0.5	0.54	0.21	0.56	0.02	0.24
German	**BiShift**	**CoordInv**	**ObjNum**	**SOMO**	**Tense**	**SentLen**	**SubjNum**	**WC**	**TreeDepth**
Attention (maxpool)	0.56	0.76	0.63	0.5	0.8	0.85	0.66	0.24	0.39
Attention (last)	0.56	0.79	0.63	0.52	0.81	0.87	0.68	0.25	0.39
RNN (maxpool)	0.57	0.8	0.64	0.51	0.82	0.68	0.69	0.28	0.37
RNN (last)	0.54	0.74	0.61	0.52	0.71	0.44	0.63	0.11	0.31
CNN (maxpool)	0.54	0.51	0.51	0.5	0.55	0.17	0.53	0.0	0.21
CNN (avg. pool)	0.54	0.5	0.53	0.5	0.57	0.21	0.54	0.01	0.23
Spanish	**BiShift**	**CoordInv**	**ObjNum**	**SOMO**	**Tense**	**SentLen**	**SubjNum**	**WC**	**TreeDepth**
Attention (maxpool)	0.57	0.72	0.69	0.51	0.85	0.82	0.73	0.25	0.44
Attention (last)	0.58	0.71	0.7	0.51	0.84	0.85	0.74	0.25	0.45
RNN (maxpool)	0.55	0.75	0.69	0.53	0.85	0.67	0.76	0.28	0.44
RNN (last)	0.55	0.7	0.65	0.52	0.75	0.54	0.68	0.12	0.36
CNN (maxpool)	0.55	0.5	0.51	0.49	0.52	0.18	0.51	0.0	0.19
CNN (avg. pool)	0.55	0.5	0.54	0.5	0.6	0.23	0.51	0.01	0.26
French	**BiShift**	**CoordInv**	**ObjNum**	**SOMO**	**Tense**	**SentLen**	**SubjNum**	**WC**	**TreeDepth**
Attention (maxpool)	0.56	0.76	0.7	0.5	0.85	0.84	0.76	0.27	0.42
Attention (last)	0.58	0.76	0.71	0.5	0.84	0.86	0.79	0.26	0.41
RNN (maxpool)	0.53	0.78	0.7	0.5	0.84	0.61	0.8	0.31	0.4
RNN (last)	0.55	0.72	0.65	0.49	0.71	0.47	0.71	0.12	0.34
CNN (maxpool)	0.55	0.52	0.49	0.51	0.5	0.17	0.51	0.0	0.2
CNN (avg. pool)	0.55	0.51	0.52	0.5	0.54	0.23	0.54	0.01	0.23
French (UN)	**BiShift**	**CoordInv**	**ObjNum**	**SOMO**	**Tense**	**SentLen**	**SubjNum**	**WC**	**TreeDepth**
Attention (maxpool)	0.57	0.74	0.7	0.5	0.82	0.83	0.76	0.27	0.42
Attention (last)	0.57	0.76	0.69	0.5	0.83	0.83	0.78	0.26	0.41
RNN (maxpool)	0.56	0.78	0.7	0.5	0.83	0.62	0.79	0.3	0.39
RNN (last)	0.55	0.73	0.65	0.5	0.68	0.47	0.71	0.13	0.34
CNN (maxpool)	0.55	0.51	0.51	0.49	0.52	0.2	0.52	0.0	0.21
CNN (avg. pool)	0.55	0.52	0.52	0.5	0.52	0.25	0.53	0.02	0.24
Russian	**BiShift**	**CoordInv**	**ObjNum**	**SOMO**	**Tense**	**SentLen**	**SubjNum**	**WC**	**TreeDepth**
Attention (maxpool)	0.58	0.66	0.56	0.52	0.74	0.82	0.6	0.2	0.35
Attention (last)	0.58	0.66	0.57	0.53	0.76	0.84	0.6	0.2	0.35
RNN (maxpool)	0.57	0.65	0.57	0.51	0.76	0.65	0.61	0.22	0.33
RNN (last)	0.57	0.57	0.56	0.52	0.68	0.45	0.59	0.11	0.3
CNN (maxpool)	0.57	0.51	0.5	0.5	0.55	0.17	0.51	0.0	0.21
CNN (avg. pool)	0.57	0.51	0.52	0.52	0.56	0.26	0.53	0.01	0.24

Table IV.4: Complete set of absolute results per probing task, per encoder, per language. For English, these numbers are for unmapped, NLI-based encoders; for all other languages, these are post-mapping numbers

Paper V

Multilingual Probing of Deep Pre-Trained Contextual Encoders

Vinit Ravishankar, Memduh Gökırmak, Lilja Øvrelid, Erik Velldal

Appears in *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, September 2019, pp. 37–47.

V

Abstract

Encoders that generate representations based on context have, in recent years, benefited from adaptations that allow for pre-training on large text corpora. Earlier work on evaluating fixed-length sentence representations has included the use of 'probing' tasks, that use diagnostic classifiers to attempt to quantify the extent to which these encoders capture specific linguistic phenomena. The principle of probing has also resulted in extended evaluations that include relatively newer word-level pre-trained encoders. We build on probing tasks established in the literature and comprehensively evaluate and analyse – from a typological perspective amongst others – multilingual variants of existing encoders on probing datasets constructed for 6 non-English languages. Specifically, we probe each layer of a multiple monolingual RNN-based ELMo models, the transformer-based BERT's cased and uncased multilingual variants, and a variant of BERT that uses a cross-lingual modelling scheme (XLM).

Contents

V.1 Introduction

Recent trends in NLP have demonstrated the utility of pre-trained deep contextual representations in numerous downstream NLP tasks, where they have almost consistently resulted in significant performance improvements. Detailed evaluations have naturally followed: these have either been follow-up works to papers describing contextual representation systems, such as Peters et al. (2018a), or novel works evaluating a broad class of encoders on a broad variety of tasks (Perone et al., 2018). This paper is an example of the latter sort; we perform a comprehensive, large-scale evaluation of what linguistic phenomena these sequential encoders capture across a diverse set of languages. This has often been referred to in the literature as *probing*; we use this terminology throughout this work.

Briefly, our goals are to probe our encoders in a multilingual setting – i.e., we use a series of probing tasks to quantify what sort of linguistic information our encoders retain, and how this information varies across language, across encoder, and across task. As such, our experiments do not attempt to attain 'state-of-the-art' results; instead, we attempt to use a comparable experimental setting across each experiment, to quantify differences between settings rather than absolute results.

In Section V.2, we describe prior work in multiple strands of research: specifically, on deep neural pre-training, on multilingualism in pre-training, and on evaluation. Section V.3 describes both the linguistic features we probe our representations for, and how we generated our probing corpus. In Section V.4, we describe and motivate our choice of encoders, as well as describe our infrastructural details. The bulk of our contribution is in Section V.5, where we describe and analyse our results. Finally, we conclude with a discussion of the implications of these results and future work in Section V.6.

V.2 Background

V.2.1 Deep pre-training

A watershed moment in NLP has been the recent innovation spree in deep pre-training; it has represented a considerable step up from shallow pre-training methods, that have been used in NLP since the introduction of contextual word embedding models such as word2vec (Mikolov et al., 2013). Whilst deep pre-training has been used in non-NLP, image-oriented tasks, where the standard paradigm is to pre-train deep convolutional networks on datasets like ImageNet (Russakovsky et al., 2014), and then fine-tune on task-specific data, their introduction to textual domains has been considerably slower, yet has been picking up rapidly in recent years.

An early paper in this theme was CoVe (McCann et al., 2017), that pre-trained contextual encoders on seq2seq machine translation models. Another earlier seminal work that addressed numerous technical issues with pre-training was Howard and Ruder's ULMFiT (2018). Not long after, the principle of deep pre-training saw widespread adoption with ELMo (Peters et al., 2018b), that consisted of several innovations over CoVe: critically, the use of an unsupervised (albeit structured) task – language modelling – for pre-training, and the use of a linear combination of all encoder layers, instead of just the top layer.

Architecturally, ELMo used two-layer bidirectional LSTMs along with character-level convolutions, to model word probabilities given the history.

With deep pre-training having been established as a valid strategy in NLP, alternative models with different underlying architectures were proposed. The OpenAI GPT (Radford et al., 2018) was one such model; instead of LSTMs, it used the *de*coder of an attention-based transformer (Vaswani et al., 2017) as its underlying *en*coder – the justification being that using the transformer's *en*coder would lead to each token having access to succeeding tokens. The GPT also achieved (then) state-of-the-art results by plugging generated fixed-length vectors into downstream classifiers.

Another system that represented a significant innovation was BERT (Devlin et al., 2018). BERT introduced a language modelling variant, dubbed *masked* language modelling, that allowed them to use transformer encoders as their underlying encoding mechanism.

V.2.2 Multilingual pre-training

Multilingual variants of pre-trained encoders that provide contextual representations for non-English languages have also been studied; there is, however, some diversity in precisely how they are generated.

Che et al. (2018) provide ELMo models (Fares et al., 2017) for 44 languages; all of these were trained on data provided as part of the CoNLL 2018 shared task on dependency parsing Universal Dependencies treebanks (Zeman et al., 2018). This makes 'multilingual' a bit of a misnomer: whilst this is the most obvious approach to multilingual *support*, these models are all *mono*lingual. This also leads to other issues downstream, such as a complete inability to deal with true multilingual phenomena like code-switching. Throughout this text, however, when not specifically referring to ELMo, our use of the term 'multilingual' is inclusive of ELMo's quasi-multilingualism.

This is contrasted with BERT's approach to (true) multilingualism, which trains a single model that can handle all languages. The authors use WordPiece, a variant of BPE (Sennrich et al., 2016), for tokenisation, using a 110K-size vocabulary, and proceed to train a single gigantic model; they perform exponentially smoothed weighting of their data to avoid biasing their model towards better-resourced languages.

Finally, XLM Lample and Conneau, 2019 is another cross-lingual encoder based on BERT that implements a number of modifications. Along with BERT's masked language modeling or Cloze task-based modelling Devlin et al., 2018; Taylor, 1953, XLM training uses another similar objective during training that the authors call translation language modeling. Here, two parallel sentences are concatenated and words masked in both source and target sentences words are predicted using context from both. The authors here also use their own implementation of BPE – FastBPE, for which they provide a vocabulary of around 120K entries. This vocabulary is shared across all of the languages and thus improves the alignment of embedded spaces, as shown in Lample et al. (2018).

V.2.3 On evaluation

Evaluation of contextual representations goes beyond merely deep representations; not too far in the past, work on evaluating shallow sentence representations was encouraged by the release of the SentEval toolkit (Conneau and Kiela, 2018), which provided an easy-to-use framework that sentence representations could be 'plugged' into, for rapid downstream evaluation on numerous tasks: these include several classification tasks, textual entailment and similarity tasks, a paraphrase detection task, and caption/image retrieval tasks. Relevant to our paper is Conneau et al.'s (2018) set of 'probing tasks', a variant on the theme of diagnostic classification (Adi et al., 2017; Belinkov et al., 2017; Hupkes et al., 2018; Shi et al., 2016), that would attempt to quantify precisely what sort of linguistic information was being retained by sentence representations. Based in part on Shi et al. (2016), Conneau et al. (2018) focus on evaluating representations for English; they provide Spearman correlations between the performance of a particular representation mechanism on being probed for specific linguistic properties, and the downstream performance on a variety of NLP tasks. Along similar lines, and contemporaneously with this work, Liu et al. (2019) probe similar deep pre-trained to the ones we do, on a set of 'sixteen diverse probing tas ks'. (Tenney et al., 2019b) probe deep pre-trained encoders for sentence structure.

On a different note, Saphra and Lopez (2019) present a CCA-based method to compare representation learning dynamics across time and models, without explicitly requiring annotated corpora.

A visible limitation of the datasets provided by these probing tasks is that most of them were created with the idea of evaluating representations built for English language data. Within the realm of evaluating *multilingual* sentence representations, **conneau_xnli:_2018** describe the XNLI dataset, a set of translations of the development and test portions of the multi-genre MultiNLI inference dataset (Williams et al., 2018). This, in a sense, is an extension of a predominantly monolingual task to the multilingual domain; the authors evaluate sentence representations derived by *mapping* non-English representations to an English representation space.

V.2.4 BERTology

Relevant to the probing theme of this paper is the sudden recent growth in papers studying precisely what is retained with the internal representations of pre-trained encoders like BERT. These include, for instance, analyses of BERT's attentions heads, such as Michel et al. (2019), where the authors prune heads, often reducing certain layers to single heads, without a significant drop in performance in certain scenarios. Clark et al. (2019) provide a per-head analysis and attempt to quantify what information each head retains; they discover that specific aspects of syntax are well-encoded per head, and find heads that correspond to certain linguistic properties, such as heads that attend to direct objects of verbs. Other papers provide analyses of BERT's layers, such as Tenney et al. (2019a), who discover that BERT's layers roughly correspond to the notion of the classical 'NLP pipeline', with lower level tasks such as tagging lower down the layer hierarchy. Hewitt and Manning (2019) define a structural probe over BERT representations, that extracts notions of syntax that correspond strongly to linguistic notions of dependency syntax.

V.3 Corpora

V.3.1 Probing

Our data consists of training, development and test splits for 9 linguistic tasks, that can broadly be grouped into surface, syntactic and semantic tasks. These are the same as the ones described in Conneau et al. (2018), with minor modifications. Due to the differences in corpus domain, we alter some of their word-frequency parameters. We also exclude the top constituent (**TopConst**) task; we noticed that Wikipedia tended to have far less diversity in sentence structure than the original Toronto Books corpus, due to the more encyclopaedic style of writing. A brief description of the tasks follows, although we urge the reader to refer to the original paper for more detailed descriptions.

1. Sentence length: In **SentLen**, sentences are divided into multiple bins based on their length; the job of the classifier is to predict the appropriate bin, creating a 6-way classification task.

2. Word count: In **WC**, we sample sentences that feature exactly one amongst a thousand mid-frequency words, and train the classifier to predict the word: this is the most 'difficult' task, in that it has the most possible classes.

3. Tree depth: The **TreeDepth** task simply asks the representation to predict the depth of the sentence's syntax tree. Unlike the original paper, we use the depth the of the dependency tree instead of the constituency tree.

4. Bigram shift: In **BiShift**, for half the sentences in the dataset, the order of words in a randomly sampled bigram is reversed. The classifier learns to predict whether or not the sentence contains a reversal.

5. Subject number: The **SubjNum** task asks the classifier to predict the number of the subject of the head verb of the sentence. Only sentences with exactly one subject (annotated with the `nsubj` relation) attached to the root verb were considered.

6. Object number: **ObjNum**, similar to the subject number task, was annotated with the number of the direct object of the head verb (annotated with the `obj` relation).

7. Coordination inversion: In **CoordInv**, two main clauses joined by a coordinating conjunction have their orders reversed, with a probability of one in two. Only sentences with exactly two top-level conjuncts are considered.

8. (Semantic) odd man out: **SOMO**, one of the more difficult tasks in the collection, replaces a randomly sampled word with another word with comparable corpus bigram frequencies.

9. Tense prediction: The **Tense** prediction asks the classifier to predict the tense of the main verb: we compare the past and present tenses.

V.3.2 Data

Languages

Our choice of languages was motivated by three factors: i) the availability of a Wikipedia large enough to extract data from; ii) the availability of a reasonable dependency parsing model, and iii) typological diversity. The former, in particular, was a bit of a restriction, since not all sentences were valid candidates for extraction per task. Our final set of languages include an additional corpus for English, as well as French, German, Spanish, Russian, Turkish and Finnish. Whilst not nearly representative of the diversity of world languages, this selection includes morphologically agglutinative, fusional and (relatively) isolating languages, and it includes two scripts, Latin and Cyrillic. The languages also represent three families (Indo-European, Turkic and Uralic).

We build our probing datasets using the relevant language's Wikipedia dump as a corpus. Our motivation for doing so was that it a freely available corpus for numerous languages, large enough to extract the sizeable corpora that we need. Specifically, we use Wikipedia dumps (dated 2019-02-01), which we process using the WikiExtractor utility[1].

The dataset, that we dub X-PROBE (Ravishankar et al., 2019), is freely available on Github[2].

Preprocessing

We use the Punkt tokeniser (Kiss and Strunk, 2006) to segment our Wikipedia dumps into discrete sentences. For Russian, which lacked a Punkt tokenisation model, we used the UDPipe (Straka and Straková, 2017) toolkit to perform segmentation.

Having segmented our data, we used the Moses (Koehn et al., 2007) tokeniser for the appropriate language, falling back to English tokenisation when unavailable.

Next, we obtained dependency parses for our sentences, again using the UDPipe toolkit's pretrained models, trained on Universal Dependencies

[1] https://github.com/attardi/wikiextractor/

[2] https://github.com/ltgoslo/xprobe

treebanks (Nivre et al., 2015). We then processed these dependency parsed corpora to extract the appropriate sentences; while in principle, each task was meant to have 120K sentences, with 100K/10K/10K training/validation/test splits, often, for the rarer linguistic phenomena, we ran out of source data, in particular with Turkish and Finnish, although to a smaller extent with Russian as well. In these situations, we ensured an equivalent split ratio.

Our use of non-gold-standard dependency parses implies inaccuracies that, in principle, would propagate to our training data. A valid counterargument, however, is that we do not rely on complete parse accuracies for all our tasks; several tasks do not require dependency or POS annotation, and the ones that do rely on a fixed subset of dependency relations, such as `nsubj` or `obj`. Having said that, we do acknowledge the divergences in parsing performance across language; unfortunately, given the substantial corpus sizes these experiments require, we could not use gold-standard parsed corpora.

V.4 Implementation

V.4.1 Encoders

We probe several popular pre-trained encoders (or, specifically, their multilingual variants). These include:

ELMo, monolingual We use Che et al.'s (2018) pre-trained monolingual ELMo models for each of our languages. Training was similar to the original English language ELMo, but allows for Unicode, and uses a sample softmax (Jean et al., 2014) to deal with large vocabularies. We probed four variants of each ELMo model - the character embeddings layer, the two LSTM layers, and an average of all three. For obtaining a fixed-length sentence representation, we use average pooling over the sequence of hidden states.

BERT We use the two multilingual variants - cased and uncased. Both variants have 12 layers, 768 hidden units, 12 heads and 110M parameters; the former includes 104 languages and fixes normalisation issues, whilst the latter includes 102 languages. For further classification, we use the first hidden state, represented by the `[CLS]` token.

XLM We probe only one variant of this encoder - i.e., the models fine-tuned on XNLI (**conneau_xnli:_2018**) data. Due to there being no XNLI data for Finnish, we do not probe our Finnish dataset with XLM. Unlike BERT, XLM uses 1024 hidden units and 8 heads.

Unfortunately, all our encoders did include Wikipedia dumps in their training data. Given that pretrained encoders tend to use as much easily accessible data as possible in pre-training, however, it is difficult to avoid using a completely unseen corpus for probing task extraction.

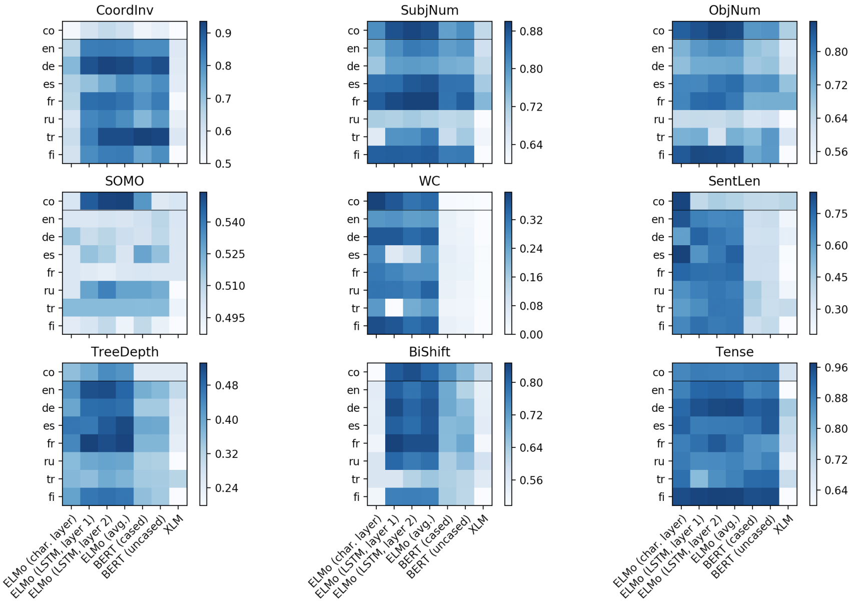


Figure V.1: Detailed results per task, per language per encoder. Each task's result heatmap has its own scale. All results mentioned in this paper refer to classification accuracies in [0.0, 1.0]. Henceforth, 'co' refers to probing results on Conneau et al.'s (2018) original corpus.

V.4.2 Implementation

Our probing procedure for each of our languages and encoders is relatively similar: we use a multi-layer perceptron based classifier to assign the appropriate class label to each input sentence. During training, the encoders remain static, with all learning restricted to the classifier. In an attempt to avoid excessively complex classifiers, and to ensure consistency across tasks and languages, we use predetermined fixed hyperparameters – specifically, a sigmoid activation function, on top of a size 50 dense layer. We use a training batch size of 32, optimised using Adam (Kingma and Ba, 2014), and train for 10 epochs, allowing for early stopping.

We implement our system using the AllenNLP toolkit (Gardner et al., 2018), which crucially provides the ability to use the appropriate tokenisation schema, along with the appropriate vocabulary, for each encoder. Training and evaluation were carried out on NVIDIA RTX 2080 Ti GPUs, with 10GiB GPU memory.

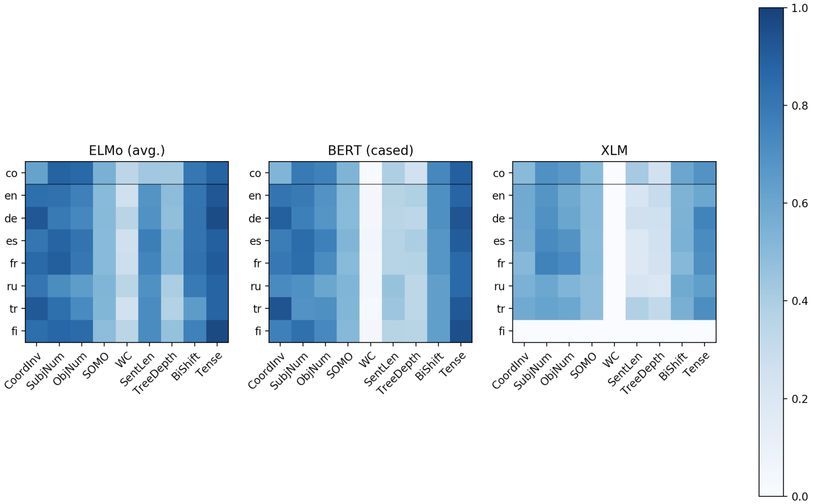


Figure V.2: Results for select encoders, per language per task. All results use the same scale, [0.0, 1.0].

V.5 Results

Due to our large experiment space, there are several dimensions along which our results can be analysed and discussed. For ease of analysis, all our figures are presented as heatmaps.

We have presented our results in two ways, for easy visualisation. The first of these is dividing them up by task, as in Figure VI.3. We present an alternative set of results for three of our encoders, in Figure V.2.

V.5.1 Encoder

An observation that instantly stands out is the significant difference in performance on WC: consistently, across every language, all our transformer-based architectures see results very close to 0. Further, whilst not instantly visible in Figure V.2, a quick look at Figure VI.3 shows that the same appears to hold (albeit to a lesser extent) for SENTLEN, TREEDEPTH and BISHIFT, all of which are either surface or syntactic phenomena. This appears to heavily imply that recurrent, sequential processing appears to retain lower level linguistic phenomena better than self-attentive mechanisms (that do not see the same drop in informativity for semantic tasks). This is perhaps a bit easy to justify with SENTLEN, which is a phenomenon that is directly proportional to recurrence depth.

The next phenomenon of interest is the difference between each of ELMo's layers. Interestingly, these do not appear to be as drastic as one would imagine,

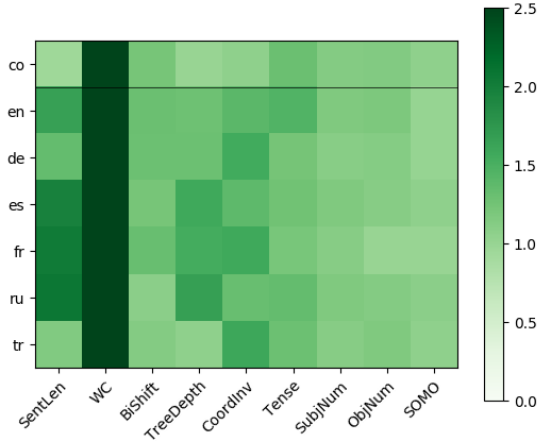


Figure V.3: BERT (cased) scores divided by the corresponding XLM scores. Tasks are ordered, from surface to syntax to semantic level tasks.

given the differences in performance on downstream tasks. The difference between raw word representations and actual contextual representations is fairly noticeable, particularly on the strongly syntactic BiShift. However, the differences between higher layers is relatively murkier, and whilst the average of the three does appear to represent some phenomena better (such as CoordInv), it isn't clear that this difference is meaningful. Notably, SentLen appears to be poorly represented in higher layers, which ties in with other analyses of ELMo (Peters et al., 2018a), that imply that higher layers are likelier to learn more semantic features.

BERT's cased variant appears to retain information slightly better than the uncased one, which is in line with the authors' descriptions of their own models.

Finally, and perhaps most interestingly, we turn our attention towards XLM. Despite being based on BERT (and indeed showing similar *patterns* in performance), XLM appears to perform a lot worse than all our other encoders on virtually every task. It is not immediately clear why: however, given that this drop in performance is visible in every language, our conjecture is that due to the translation-based modelling employed by XLM, the encoder does indeed succeed at learning language-independent representations, or 'universal' representations. However, this universality comes at a cost: in an attempt to adequately represent a variety of typologically diverse languages, XLM appears to lose its ability to retain *specific* linguistic phenomena pertaining to specific languages; in a sense, it is incapable of building a representation for a language that adequately captures a specific phenomenon in that language *and no other*. This follows intuitively from the method used training on the TLM objective: the authors concatenate aligned parallel sentences and predict masked words in the source *and* the target sentence, using context from both sentences at the same time to predict each masked word. This is likely to have had a detrimental effect

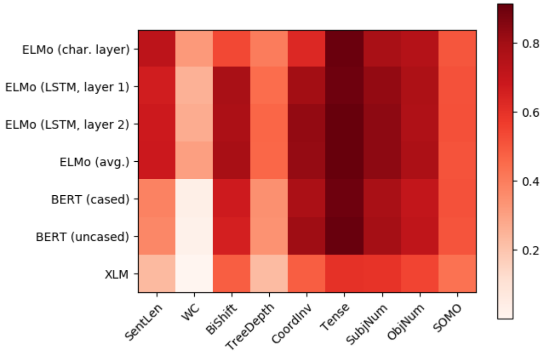


Figure V.4: Linguistic information retained per encoder, per task; scores are averaged over language.

on XLM's ability to retain characteristics specific to each language. In Figure V.3, we show the relative performance of BERT and XLM per probing task. There is a clear trend towards BERT's enhanced retention of linguistic features being less prominent for the more semantic tasks, which fits our hypothesis, as semantics are likelier to hold cross-linguistically.

A point to be made here is that despite SubjNum, ObjNum and Tense being classified as semantic tasks, it isn't clear that they are truly being probed for semantic information: all three phenomena tend to be visible with morphological marking. This gives us an alternative justification for XLM's relative improvement in retention: XLM is likely capable of storing each language's individual morphological information in different internal subspaces de, as each language is likely to reflect morphology purely orthographically, and in mutually exclusive ways.

Our observations on the differences between encoders are also easily visible in Figure V.4, where multiple 'belts' of varying performances emerge.

V.5.2 Language

To motivate one of the main focuses of this paper – our analysis of our results along linguistic lines – we present Figure V.5, which displays what one might call the net 'informativity' of an encoder, i.e. an average of how much information each encoder retains averaged over tasks. The most noticeable effect here is the drop in informativity for Russian and Turkish. While this is perhaps understandable for Turkish – which has smaller probing corpora, and a less reliable Wikipedia than the other languages – Russian's opaqueness cannot be as easily explained away, particularly when contrasted with Finnish, which tends to have fewer resources.

We further introduce Figure V.6, which displays the averaged results of three systems – ELMo's multilayer variant, BERT's cased variant and (absent for Finnish) XLM. Most linguistic differences appear to be clustered in the semantic

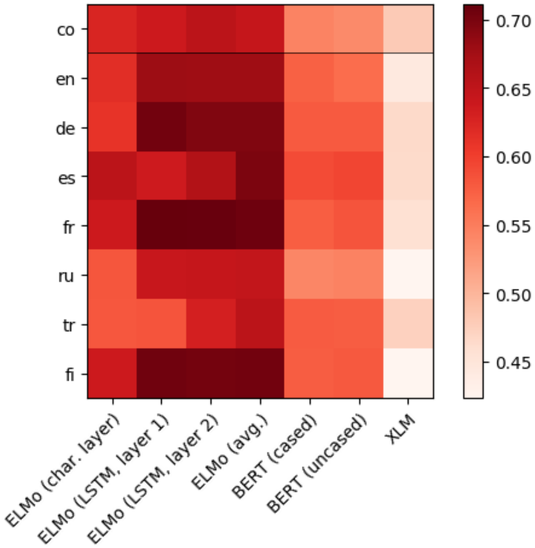


Figure V.5: Net encoder 'informativity' per language; results averaged over all tasks.

part of this heatmap. There are numerous possible factors that could explain these divergences, not the least of which is the actual probing corpus itself: however, we attempt to provide a justification, from a typological perspective, for some of these results.

When averaged across encoders, the TENSE task stands out as fairly easy to probe for all languages. It thus seems that information about verbal temporal properties is retained in the sentence representation. For the tasks of subject and object number, however, we observe clear differences between the languages. Here, French and Spanish appear to be somewhat easier to probe than other languages. We hypothesise that this is due to both languages marking nominal number, not just with verb agreement, but also with plural articles, resulting in representations that are more informative regarding number. Contrast this with English and German, which either do not have plural articles, or have plural articles that morphologically overlap with non-plural forms, or with Russian, that tends to avoid articles in general.

Other interesting observations are German's relative ability at retaining information on COORDINV and TENSE, as well as Finnish's extraordinarily high performance on TENSE. Further, SENTLEN appears to be retained better, counter-intuitively, in Russian, Turkish and Finnish; a brief look at Figure VI.3 shows that, interestingly, this is likely due to BERT.

Finally, we note that our results do not seem to indicate that English is somehow better represented in our multilingual systems, nor does it appear to perform significantly better than other languages in general, indicating that

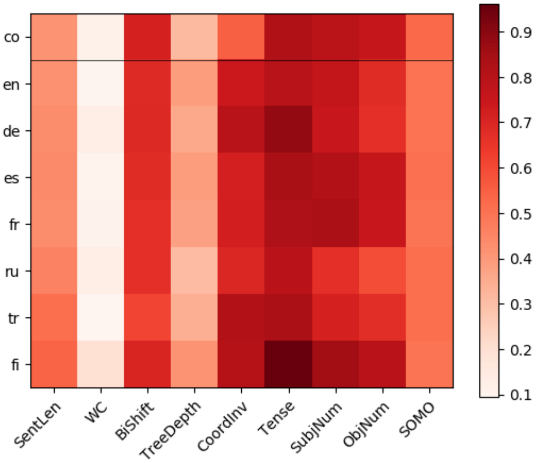


Figure V.6: Linguistic insight per language per task, averaged over one variant of every encoder: multi-layer ELMo, cased BERT, and XLM (bar Finnish).

none of our models are 'learning' English first and then adapting to other languages.

V.5.3 Task

From a monolingual perspective, most of what needs to be said regarding the choice of probing tasks has already been said in the original (Conneau et al., 2018). There are however several differences, induced both by our modifications to the original framework, and by our corpus's multilingualism.

The first of these is the apparently consistent differences in performance on certain tasks which include, amongst others, COORDINV, where our variant appears to be more easily retained than the original. This can be explained away by minor issues we faced during implementation, using dependency trees instead of constituency trees. Due to more complicated representation of conjuncts in UD-style dependency trees, some of our sentences had issues with using the appropriate casing after swapping conjuncts, as well as ensuring consistent punctuation. While we attempted to avoid these by writing filtering rules, these were imperfect, and it is likely that stray punctuation and the like might have informed our representations about the conjuncts being swapped, in some instances.

Another task with minor differences is our implementation of SOMO; we attribute this to not being able to accurately reproduce Conneau et al.'s (2018) modified corpus-frequency range (40-400) to adequately fit all our corpora.

We note that there do not appear to be significant differences in the TREEDEPTH task, despite our using dependency trees instead of constituency, and despite our tree depth/sentence length de-correlation procedure being markedly simpler.

V.6 Discussion

V.6.1 Implications

Having elaborated our results, it becomes crucial to contextualise their importance. 'Probing' an encoder, or more correctly, using diagnostic classifiers to attempt to *quantify* what information an encoder stores, appears to be a reasonable approach to *qualifying* this information. However, there has been some critique of this approach. To paraphrase Saphra and Lopez (2019), the architecture of a diagnostic classifier does affect the performance of a probing task; further, lower layers of encoders may represent information in ways designed to be picked up on by their own higher layers; this might prove difficult for simple classifiers to truly probe.

This is an excellent critique of the principle using *absolute* probing performance, or *absolute* numbers representing performance on an abstract insight task, as a yardstick. Critically, this work is focussed, both practically and in principle, on elucidating *relative* results, in a wide space of languages and encoders. The relative underparameterisation of the classifier and the use of one constant set of hyperparameters across experiments is an attempt to minimise the *relative* interference of the classifier. i.e., our goal is to keep the classifier's interference – its lens – as consistent as possible.

V.6.2 Future work

One potential strand of research relates directly to the tasks themselves: our choice of tasks was fairly restrictive, and does not include many tasks that are truly *semantic*, which does not provide us with enough information to draw conclusions similar to Liu et al. (2019), which is that pretrained models encode stronger syntax than semantics. An obvious goal, therefore, is the more careful design of tasks, particularly within a multilingual context: the tasks proposed by Liu et al. (2019) and Tenney et al. (2019b) are not strictly easy to motivate cross-linguistically due to the burden of annotation. This could include more semantic-level probing by means of existing cross-lingual semantic resources, such as the Parallel Meaning Bank (Abzianidze et al., 2017).

References

Abzianidze, L. et al. (2017). "The Parallel Meaning Bank: Towards a Multilingual Corpus of Translations Annotated with Compositional Meaning Representations". In: *arXiv:1702.03964 [cs]*. arXiv: 1702.03964.

Adi, Y. et al. (2017). "Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Belinkov, Y. et al. (2017). “What do Neural Machine Translation Models Learn about Morphology?” en. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics.

Che, W. et al. (2018). “Towards Better UD Parsing: Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation”. In: *arXiv:1807.03121 [cs]*. arXiv: 1807.03121.

Clark, K. et al. (2019). “What Does BERT Look At? An Analysis of BERT’s Attention”. In: *arXiv preprint arXiv:1906.04341*.

Conneau, A. and Kiela, D. (2018). “SentEval: An Evaluation Toolkit for Universal Sentence Representations”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Conneau, A. et al. (2018). “What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties”. en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Devlin, J. et al. (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805 [cs]*. arXiv: 1810.04805.

Fares, M. et al. (2017). “Word vectors, reuse, and replicability: Towards a community repository of large-text resources”. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics.

Gardner, M. et al. (2018). “AllenNLP: A Deep Semantic Natural Language Processing Platform”. In: *arXiv:1803.07640 [cs]*. arXiv: 1803.07640.

Hewitt, J. and Manning, C. D. (2019). “A structural probe for finding syntax in word representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Howard, J. and Ruder, S. (2018). “Universal Language Model Fine-tuning for Text Classification”. In: *arXiv:1801.06146 [cs, stat]*. arXiv: 1801.06146.

Hupkes, D. et al. (2018). “Visualisation and ’diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure”. In: *Journal of Artificial Intelligence Research* vol. 61.

Jean, S. et al. (2014). “On Using Very Large Target Vocabulary for Neural Machine Translation”. In: *arXiv:1412.2007 [cs]*. arXiv: 1412.2007.

Kingma, D. P. and Ba, J. (2014). “Adam: A Method for Stochastic Optimization”. In: *arXiv:1412.6980 [cs]*. arXiv: 1412.6980.

Kiss, T. and Strunk, J. (2006). “Unsupervised multilingual sentence boundary detection”. In: *Computational Linguistics* vol. 32, no. 4.

Koehn, P. et al. (2007). “Moses: Open source toolkit for statistical machine translation”. In: *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*.

Lample, G. and Conneau, A. (2019). “Cross-Lingual Language Model Pretraining”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Lample, G. et al. (2018). “Unsupervised Machine Translation Using Monolingual Corpora Only”. In: *International Conference on Learning Representations*.

Liu, N. F. et al. (2019). “Linguistic Knowledge and Transferability of Contextual Representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

McCann, B. et al. (2017). “Learned in Translation: Contextualized Word Vectors”. In: NIPS’17.

Michel, P. et al. (2019). “Are Sixteen Heads Really Better than One?” In: *arXiv preprint arXiv:1905.10650*.

Mikolov, T. et al. (2013). “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems 26*. Ed. by Burges, C. J. C. et al. Curran Associates, Inc.

Nivre, J. et al. (2015). *Universal Dependencies 1.2*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Perone, C. S. et al. (2018). “Evaluation of sentence embeddings in downstream and linguistic probing tasks”. In: *arXiv:1806.06259 [cs]*. arXiv: 1806.06259.

Peters, M. et al. (2018a). “Dissecting Contextual Word Embeddings: Architecture and Representation”. en. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Peters, M. E. et al. (2018b). “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.

Radford, A. et al. (2018). “Improving Language Understanding by Generative Pre-Training”. en. In.

Ravishankar, V. et al. (2019). “Probing Multilingal Sentence Representations With X-Probe”. In: *arXiv preprint arXiv:1906.05061*.

Russakovsky, O. et al. (2014). “ImageNet Large Scale Visual Recognition Challenge”. In: *arXiv:1409.0575 [cs]*. arXiv: 1409.0575.

Saphra, N. and Lopez, A. (2019). “Understanding Learning Dynamics Of Language Models with SVCCA”. 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.

Sennrich, R. et al. (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.

Shi, X. et al. (2016). “Does String-Based Neural MT Learn Source Syntax?” en. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics.

Straka, M. and Straková, J. (2017). "Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe". In: *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver, Canada: Association for Computational Linguistics.

Taylor, W. L. (1953). ""Cloze procedure": A new tool for measuring readability". In: *Journalism Bulletin* vol. 30, no. 4.

Tenney, I. et al. (2019a). "BERT rediscovers the classical NLP pipeline". In: *arXiv preprint arXiv:1905.05950*.

Tenney, I. et al. (2019b). "What do you learn from context? Probing for sentence structure in contextualized word representations". In: *International Conference on Learning Representations*.

Vaswani, A. et al. (2017). "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al. Vol. 30. Curran Associates, Inc.

Williams, A. et al. (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". en. 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.

Zeman, D. et al. (2018). "CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies". In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*.

Appendix V.A Detailed results

English	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.65	0.76	0.71	0.50	0.27	0.80	0.43	0.55	0.88
ELMo (LSTM, layer 1)	0.84	0.81	0.76	0.50	0.27	0.69	0.51	0.81	0.92
ELMo (LSTM, layer 2)	0.84	0.84	0.77	0.50	0.25	0.67	0.52	0.78	0.93
ELMo (lin. comb.)	0.84	0.83	0.77	0.50	0.26	0.68	0.49	0.81	0.92
BERT (cased)	0.81	0.79	0.69	0.50	0.04	0.36	0.39	0.70	0.88
BERT (uncased)	0.81	0.80	0.68	0.51	0.02	0.37	0.37	0.63	0.89
XLM	0.58	0.68	0.58	0.50	0.00	0.22	0.30	0.54	0.60
German	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.72	0.73	0.69	0.52	0.37	0.60	0.40	0.54	0.92
ELMo (LSTM, layer 1)	0.91	0.79	0.73	0.51	0.36	0.77	0.48	0.83	0.95
ELMo (LSTM, layer 2)	0.93	0.80	0.73	0.51	0.34	0.72	0.48	0.80	0.96
ELMo (lin. comb.)	0.92	0.79	0.74	0.51	0.36	0.69	0.48	0.82	0.96
BERT (cased)	0.90	0.77	0.68	0.50	0.04	0.35	0.34	0.70	0.93
BERT (uncased)	0.92	0.77	0.69	0.51	0.03	0.36	0.34	0.67	0.94
XLM	0.57	0.70	0.60	0.50	0.00	0.26	0.26	0.54	0.75
Spanish	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.67	0.85	0.78	0.50	0.29	0.85	0.47	0.54	0.93
ELMo (LSTM, layer 1)	0.71	0.85	0.78	0.52	0.07	0.64	0.46	0.81	0.89
ELMo (LSTM, layer 2)	0.75	0.88	0.81	0.51	0.11	0.71	0.50	0.79	0.89
ELMo (lin. comb.)	0.81	0.88	0.82	0.50	0.26	0.78	0.52	0.82	0.89
BERT (cased)	0.78	0.85	0.77	0.53	0.05	0.36	0.40	0.68	0.91
BERT (uncased)	0.81	0.85	0.77	0.52	0.04	0.37	0.39	0.68	0.94
XLM	0.56	0.72	0.69	0.50	0.00	0.19	0.25	0.55	0.72
French	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.66	0.87	0.79	0.50	0.32	0.76	0.44	0.52	0.89
ELMo (LSTM, layer 1)	0.87	0.89	0.82	0.50	0.30	0.74	0.53	0.85	0.91
ELMo (LSTM, layer 2)	0.87	0.90	0.83	0.50	0.28	0.74	0.52	0.83	0.94
ELMo (lin. comb.)	0.86	0.90	0.80	0.50	0.28	0.75	0.53	0.83	0.91
BERT (cased)	0.80	0.84	0.72	0.50	0.05	0.37	0.37	0.67	0.86
BERT (uncased)	0.84	0.87	0.72	0.50	0.04	0.36	0.37	0.71	0.85
XLM	0.51	0.76	0.72	0.50	0.00	0.18	0.24	0.51	0.70
Russian	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.60	0.72	0.63	0.50	0.32	0.65	0.36	0.57	0.89
ELMo (LSTM, layer 1)	0.82	0.72	0.64	0.53	0.32	0.69	0.40	0.80	0.87
ELMo (LSTM, layer 2)	0.84	0.73	0.63	0.54	0.31	0.72	0.40	0.77	0.87
ELMo (lin. comb.)	0.81	0.71	0.65	0.53	0.35	0.70	0.40	0.79	0.88
BERT (cased)	0.72	0.70	0.61	0.53	0.04	0.46	0.33	0.64	0.86
BERT (uncased)	0.79	0.71	0.62	0.53	0.03	0.38	0.33	0.67	0.88
XLM	0.55	0.60	0.53	0.49	0.00	0.22	0.20	0.58	0.64
Turkish	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.63	0.65	0.72	0.52	0.26	0.61	0.37	0.57	0.91
ELMo (LSTM, layer 1)	0.84	0.81	0.72	0.52	0.00	0.65	0.36	0.57	0.79
ELMo (LSTM, layer 2)	0.92	0.81	0.61	0.52	0.23	0.72	0.39	0.63	0.86
ELMo (lin. comb.)	0.92	0.84	0.72	0.52	0.26	0.72	0.38	0.65	0.88
BERT (cased)	0.94	0.69	0.70	0.52	0.03	0.44	0.34	0.63	0.92
BERT (uncased)	0.93	0.73	0.76	0.52	0.02	0.37	0.34	0.62	0.92
XLM	0.58	0.62	0.60	0.49	0.00	0.38	0.32	0.56	0.71
Finnish	**CoordInv**	**SubjNum**	**ObjNum**	**SOMO**	**WC**	**SentLen**	**TreeDepth**	**BiShift**	**Tense**
ELMo (char. layer)	0.60	0.87	0.84	0.50	0.38	0.67	0.41	0.51	0.96
ELMo (LSTM, layer 1)	0.81	0.87	0.86	0.50	0.37	0.74	0.47	0.77	0.97
ELMo (LSTM, layer 2)	0.84	0.87	0.86	0.51	0.33	0.71	0.47	0.77	0.97
ELMo (lin. comb.)	0.84	0.87	0.85	0.49	0.35	0.72	0.47	0.76	0.97
BERT (cased)	0.77	0.84	0.73	0.51	0.04	0.36	0.36	0.64	0.95
BERT (uncased)	0.81	0.84	0.76	0.49	0.03	0.39	0.34	0.61	0.96
XLM	-	-	-	-	-	-	-	-	-

Table V.1: Detailed table with probing results

Appendix V.B Parser results

Language	UPOS	Feats	AllTags	Lemmas	UAS	LAS
English	93.50	94.44	91.48	96.10	80.34	77.25
German	90.72	80.46	76.26	95.38	74.15	68.61
Spanish	95.54	96.10	93.70	95.89	85.32	81.95
French	95.49	95.42	94.26	96.59	84.09	80.50
Russian	94.69	84.17	82.61	74.91	80.94	76.15
Turkish	91.51	86.70	84.60	89.60	60.78	53.78
Finnish	94.49	91.42	90.35	86.49	80.74	77.26

Table V.2: UDPipe v1.2 parsing and tagging accuracies; UAS and LAS are unlabelled and labelled attachement scores respectively

Paper VI

Do Neural Language Models Show Preferences for Syntactic Formalisms?

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, Joakim Nivre

Appears in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 2020, pp. 4077–4091. DOI: 10.18653/v1/2020.acl-main.375.

Abstract

Recent work on the interpretability of deep neural language models has concluded that many properties of natural language syntax are encoded in their representational spaces. However, such studies often suffer from limited scope by focusing on a single language and a single linguistic formalism. In this study, we aim to investigate the extent to which the semblance of syntactic structure captured by language models adheres to a surface-syntactic or deep syntactic style of analysis, and whether the patterns are consistent across different languages. We apply a probe for extracting directed dependency trees to BERT and ELMo models trained on 13 different languages, probing for two different syntactic annotation styles: Universal Dependencies (UD), prioritizing deep syntactic relations, and Surface-Syntactic Universal Dependencies (SUD), focusing on surface structure. We find that both models exhibit a preference for UD over SUD — with interesting variations across languages and layers — and that the strength of this preference is correlated with differences in tree shape.

Contents

VI.1 Introduction

Recent work on interpretability in NLP has led to the consensus that deep neural language models trained on large, unannotated datasets manage to encode various aspects of syntax as a byproduct of the training objective. Probing approaches applied to models like ELMo (Peters et al., 2018a) and BERT (Devlin et al., 2019) have demonstrated that one can decode various linguistic properties such as part-of-speech categories, dependency relations, and named-entity types directly from the internal hidden states of a pretrained model (Peters et al., 2018b; Tenney et al., 2019b). Another line of work has tried to tie cognitive measurements or theories of human linguistic processing to the machinations of language models, often establishing strong parallels between the two (Abnar et al., 2019; Gauthier and Levy, 2019; Prasad et al., 2019).

As is the case for NLP in general, English has served as the de facto testing ground for much of this work, with other languages often appearing as an afterthought. However, despite its ubiquity in the NLP literature, English is generally considered to be atypical across many typological dimensions. Furthermore, the tendency of interpreting NLP models with respect to existing, canonical datasets often comes with the danger of conflating the theory-driven annotation therein with scientific fact. One can observe this to an extent with the Universal Dependencies (UD) project Nivre et al., 2016, which aims to collect syntactic annotation for a large number of languages. Many interpretability studies have taken UD as a basis for training and evaluating probes, but often fail to mention that UD, like all annotation schemes, is built upon specific theoretical assumptions, which may not be universally accepted.

Our research questions start from these concerns. When probing language models for syntactic dependency structure, is UD — with its emphasis on syntactic relations between content words — really the best fit? Or is the representational structure of such models better explained by a scheme that is more oriented towards surface structure, such as the recently proposed Surface-Syntactic Universal Dependencies (SUD) Gerdes et al., 2018? And are these patterns consistent across typologically different languages? To explore these questions, we fit the structural probe of Hewitt and Manning (2019) on pretrained BERT and ELMo representations, supervised by UD/SUD treebanks for 13 languages, and extract directed dependency trees. We then conduct an extensive error analysis of the resulting probed parses, in an attempt to qualify our findings. Our main contributions are the following:

1. A simple algorithm for deriving directed trees from the disjoint distance and depth probes introduced by Hewitt and Manning (2019).
2. A multilingual analysis of the probe's performance across 13 different treebanks.

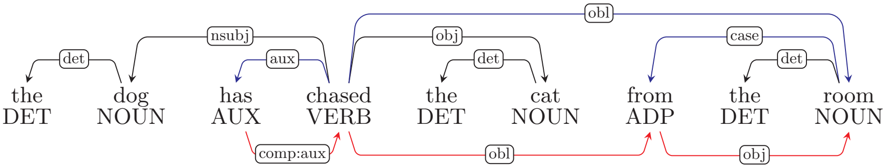


Figure VI.1: Simplified UD and SUD annotation for an English sentence.

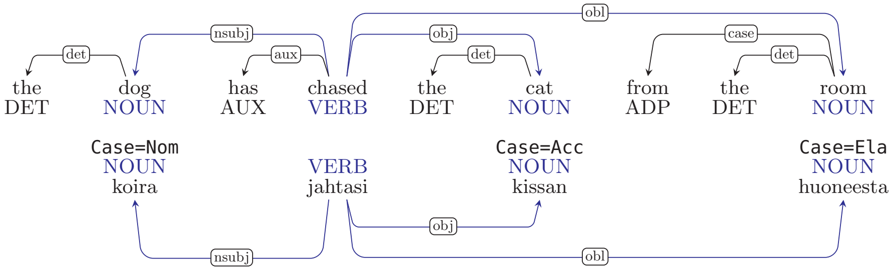


Figure VI.2: Simplified UD and SUD annotation for an English sentence.

3. An analysis showing that the syntactic information encoded by BERT and ELMo fit UD better than SUD for most languages.

VI.2 Related Work

There has been a considerable amount of recent work attempting to understand what aspects of natural language pre-trained encoders learn. The classic formulation of these probing experiments is in the form of diagnostic classification (Belinkov et al., 2017; Conneau et al., 2018; Ettinger et al., 2016; Hupkes et al., 2018), which attempts to unearth underlying linguistic properties by fitting relatively underparameterised linear models over representations generated by an encoder. These methods have also faced recent critique, for example, concerning the lack of transparency in the classifers' ability to *extract* meaningful information, as opposed to *learning* it. Alternative paradigms for interpretability have therefore been proposed, such as correlation-based methods (Chrupała and Alishahi, 2019; Kornblith et al., 2019; Raghu et al., 2017; Saphra and Lopez, 2018). However, this critique does not invalidate diagnostic classification: indeed, more recent work (Hewitt and Liang, 2019) describes methods to show the empirical validity of certain probes, via control tasks.

Among probing studies specifically pertinent to our paper, Blevins et al. (2018) demonstrate that deep RNNs are capable of encoding syntax given a variety of pre-training tasks, including language modeling. Peters et al. (2018b) demonstrate that, regardless of encoder (recurrent, convolutional, or self-attentive), biLM-based pre-training results in similar high-quality representations that implicitly encode a variety of linguistic phenomena, layer by layer. Similarly, Tenney et al. (2019a) employ the 'edge probing' approach of Tenney et al. (2019b) to demonstrate that BERT implicitly learns the 'classical NLP pipeline', with lower-level linguistic tasks encoded in lower layers and more complex phenomena in higher layers, and dependency syntax in layer 5–6. Finally, Hewitt and Manning (2019) describe a syntactic probe for extracting aspects of dependency syntax from pre-trained representations, which we describe in Section VI.4.

VI.3 Aspects of Syntax

Syntax studies how natural language encodes meaning using expressive devices such as word order, case marking and agreement. Some approaches emphasize the formal side and primarily try to account for the distribution of linguistic forms. Other frameworks focus on the functional side to capture the interface to semantics. And some theories use multiple representations to account for both perspectives, such as c-structure and f-structure in LFG (Bresnan, 2000; Kaplan and Bresnan, 1982) or surface-syntactic and deep syntactic representations in Meaning-Text Theory (Mel'čuk, 1988).

When asking whether neural language models learn syntax, it is therefore relevant to ask which aspects of syntax we are concerned with. This is especially important if we probe the models by trying to extract syntactic representations, since these representations may be based on different theoretical perspectives. As a first step in this direction, we explore two different dependency-based syntactic representations, for which annotations are available in multiple languages. The first is Universal Dependencies (UD) (Nivre et al., 2016), a framework for cross-linguistically consistent morpho-syntactic annotation, which prioritizes direct grammatical relations between content words. These relations tend to be more parallel across languages that use different surface features to encode the relations. The second is Surface-Syntactic Universal Dependencies (SUD) (Gerdes et al., 2018), a recently proposed alternative to UD, which gives more prominence to function words in order to capture variations in surface structure across languages.

Figure VI.2 contrasts the two frameworks by showing how they annotate an English sentence. While the two annotations agree on most syntactic relations (in black), including the analysis of core grammatical relations like subject (nsubj[1]) and object (obj), they differ in the analysis of auxiliaries and prepositional phrases. The UD annotation (in blue) treats the main verb *chased* as the root of the clause, while the SUD annotation (in red) assigns this role to the auxiliary *has*. The UD annotation has a direct oblique relation between *chased* and *room*, treating the preposition *from* as a case marker, while the SUD annotation has an oblique relation between *chased* and *from*, analyzing *room* as the object of *from*. The purpose of the UD style of annotation is to increase the probability of the root and oblique relations being parallel in other languages that use morphology (or nothing at all) to encode the information expressed by

[1] UD uses the *nsubj* relation, for *nominal* subject, while SUD uses a more general *subj* relation.

auxiliaries and adpositions. SUD is instead designed to bring out differences in surface structure in such cases.

The different treatment of function words affects not only adpositions (prepositions and postpositions) and auxiliaries (including copulas), but also subordinating conjunctions and infinitive markers. Because of these systematic differences, dependency trees in UD tend to have longer average dependency length and smaller height[2] than in SUD.

VI.4 Probing Model

To conduct our experiments, we make use of the structural probe proposed by Hewitt and Manning (2019), which is made up of two complementary components — distance and depth. The former is an intuitive proxy for the notion of two words being connected by a dependency: any two words w_i, w_j in a tree T are neighbors if their respective distance in the tree amounts to $d_T(w_i, w_j) = 1$. This metric can theoretically be applied to the vector space of any pretrained neural language model sentence encoding, which ouputs a set of vectors $S = \mathbf{h}_1, ..., \mathbf{h}_n$ for a sentence. In practice, however, the distance between any two vectors $\{\mathbf{h}_i, \mathbf{h}_j\} \in S$ will not be directly comparable to their distance in a corresponding syntactic tree T, because the model does not encode syntax in isolation. To resolve this, Hewitt and Manning (2019) propose to learn a linear transformation matrix B, such that $d_B(\mathbf{h}_i, \mathbf{h}_j)$ extracts the distance between any two words w_i, w_j in a parse tree. For an annotated corpus of L sentences, the distance probe can be learned via gradient descent as follows:

$$\min_B \sum_{l=1}^{L} \frac{1}{|n^l|^2} \sum_{i,j} |d_{T^l}(w_i^l, w_j^l) - d_B(\mathbf{h}_i^l, \mathbf{h}_j^l)^2|$$

where $|n^l|$ is the length of sentence l, normalized by the number $|n^l|^2$ of word pairs, and $d_{T^l}(w_i^l, w_j^l)$ is the distance of words w_i^l and w_j^l in the gold tree.

While the distance probe can predict which words enter into dependencies with one another, it is insufficient for predicting which word is the head. To resolve this, Hewitt and Manning (2019) employ a separate probe for tree depth,[3] where they make a similar assumption as they do for distance: a given (square) vector L2 norm $||\mathbf{h}_i^2||$ is analogous to w_i's depth in a tree T. A linear transformation matrix B can therefore be learned in a similar way:

$$\min_B \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i}^{n} (||w_i^l|| - ||B\mathbf{h}_i^l||^2)$$

where $||w_i^l||$ is the depth of a w_i^l in the gold tree.

[2] The *height* of a tree is the length of the longest path from the root to a leaf (sometimes referred to as *depth*).

[3] The *depth* of a node is the length of the path from the root.

Algorithm 1 Invoke CLE for sentence $S = w_{1,n}$
given distance matrix E and depth vector D

```
procedure InvokeCLE($E, D$)
    $N \leftarrow |S| + 1$
    $M \leftarrow$ Init($shape = (N, N), value = -\infty$)
    for $(w_i, w_j) \in E$ do
        if $D(w_i) < D(w_j)$ then
            $M(w_i, w_j) \leftarrow -E(w_i, w_j)$
    $root \leftarrow \mathbf{argmin}_i D(w_i)$
    $M(0, w_{root}) \leftarrow 0$
    return CLE($M$)
end procedure
```

To be able to score probed trees (against UD and SUD gold trees) using the standard metric of unlabeled attachment score (UAS), we need to derive a rooted directed dependency tree from the information provided by the distance and depth probes. Algorithm 1 outlines a simple method to retrieve a well-formed tree with the help of the Chu-Liu-Edmonds maximum spanning tree algorithm (Chu and Liu, 1965; McDonald et al., 2005). Essentially, in a sentence $S = w_1 \dots w_n$, for every pair of nodes (w_i, w_j) with an estimated distance of d between them, if w_i has smaller depth than w_j, we set the weight of the arc (w_i, w_j) to $-d$; otherwise, we set the weight to $-\infty$. This is effectively a mapping from distances to scores, with larger distances resulting in lower arc scores from the parent to the child, and infinitely low scores from the child to the parent. We also add a pseudo-root w_0 (essential for decoding), which has a single arc pointing to the shallowest node (weighted 0). We use the AllenNLP (Gardner et al., 2018) implementation of the Chu-Liu/Edmonds' algorithm.

VI.5 Experimental Design

In order to evaluate the extent to which a given model's representational space fits either annotation framework, we fit the structural probe on the model, layer by layer, using UD and SUD treebanks for supervision, and compute UAS over each treebank's test set as a proxy for a given layer's goodness-of-fit.

Language and Treebank Selection We reuse the sample of Kulmizev et al. (2019), which comprises 13 languages from different language families, with different morphological complexity, and with different scripts. We use treebanks from UD v2.4 Nivre et al., 2019 and their conversions into SUD.[4] Table VI.1 shows background statistics for the treebanks, including the percentage of adpositions (ADP) and auxiliaries (AUX), two important function word categories that are treated differently by UD and SUD. A direct comparison

[4] https://surfacesyntacticud.github.io/data/

Language	Code	Treebank	# Sents	%ADP	%AUX	%ContRel		Dep Len		Height	
						UD	SUD	UD	SUD	UD	SUD
Arabic	arb	PADT	6075	15	1	37	24	4.17	3.92	7.20	9.82
Chinese	cmn	GSD	3997	5	3	37	30	3.72	3.74	4.30	6.56
English	eng	EWT	12543	8	6	20	12	3.13	2.94	3.48	5.11
Basque	eus	BDT	5396	2	13	34	25	2.99	2.90	3.49	4.18
Finnish	fin	TDT	12217	2	7	35	30	2.98	2.91	3.42	4.22
Hebrew	heb	HTB	5241	14	2	28	14	3.76	3.53	5.07	7.30
Hindi	hin	HDTB	13304	22	9	26	10	3.44	3.05	4.25	7.41
Italian	ita	ISDT	13121	14	5	21	8	3.30	3.12	4.21	6.28
Japanese	jap	GSD	7125	25	14	31	10	2.49	2.08	4.40	8.18
Korean	kor	GSD	4400	2	0	58	57	2.20	2.17	3.86	4.07
Russian	rus	SynTagRus	48814	10	1	31	22	3.28	3.13	4.21	5.24
Swedish	swe	Talbanken	4303	12	5	29	17	3.14	2.98	3.50	5.02
Turkish	tur	IMST	3664	3	2	33	30	2.21	2.12	3.01	3.37
Average	-	-	10784.62	12	5	32	22	3.14	3.00	4.20	5.91

Table VI.1: Treebank statistics: number of sentences (# Sents) and percentage of adpositions (ADP) and auxiliaries (AUX). Comparison of UD and SUD: percentage of direct relations involving only nouns and/or verbs (ContRel); average dependency length (DepLen) and average tree height (Height). Language codes are ISO 639-3.

of the UD and SUD representations shows that, as expected, UD has a higher percentage of relations directly connecting nouns and verbs (ContRel), higher average dependency length (DepLen) and lower average tree height (Height). However, the magnitude of the difference varies greatly across languages.[5]

Models We evaluate two pretrained language models: BERT (Devlin et al., 2019) and ELMo (Peters et al., 2018a). For BERT, we use the pretrained `multilingual-bert-cased` model provided by Google.[6] The model is trained on the concatenation of WikiDumps for the top 104 languages with the largest Wikipedias and features a 12-layer Transformer with 768 hidden units and 12 self-attention heads. For ELMo, we make use of the pretrained monolingual models made available by Che et al. (2018). These models are trained on 20 million words randomly sampled from the concatenation of WikiDump and CommonCrawl datasets for 44 different languages, including our 13 languages. Each model features a character-based word embedding layer, as well as 2 bi-LSTM layers, each of which is 1024-dimensions wide.

Though we fit the probe on all layers of each model separately, we also learn a weighted average over each full model:

$$\mathbf{model}_i = \sum_{j=0}^{L} s_j \mathbf{h}_{i,j}$$

[5] For Chinese, UD actually has slightly lower average dependency length than SUD.

[6] https://github.com/google-research/bert

where s_j is a learned parameter, $\mathbf{h}_{i,j}$ is the encoding of word i at layer j, and L is the number of layers. We surmise that, in addition to visualizing the probes' fit across layers, this approach will give us a more general notion of how well either model aligns with the respective frameworks. We refer to this representation as the 13th BERT layer and the 3rd ELMo layer. When determining the dimensionality of the transformation matrix (i.e. probe rank), we defer to each respective encoder's hidden layer sizes. However, preliminary experiments indicated that probing accuracy was stable across ranks of decreasing sizes.

It is important to note that by *probe* we henceforth refer to the algorithm that combines both distance and depth probes to return a valid tree. One could argue that, per recent insights in the interpretability literature (e.g. Hewitt and Liang, 2019), this model is too expressive in that it combines supervision from two different sources. We do not consider this a problem, as the two probes are trained separately and offer views into two different abstract properties of the dependency tree. As such, we do not optimize for UAS directly.

VI.6 Results and Discussion

Figure VI.3 displays the UAS after fitting the structural probes on BERT and ELMo, per language and layer. What is perhaps most noticeable is that, while BERT can achieve accuracies upwards of 79 UAS on some languages, ELMo fares consistently worse, maxing out at 65 for Hindi at layer 2. The most likely explanation for this is that the ELMo models are smaller than the multilingual BERT's 12-layer Transformer-based architecture, which was trained on orders of magnitude more data (albeit multilingually).

In general, we find that the probing performance is stable across languages, where layers 7–8 fare the best for BERT and layer 2 for ELMo.[7] This contrasts with prior observations Tenney et al., 2019a, as the syntactic 'center of gravity' is placed higher in each model's hierarchy. However, computing a weighted average over layers tends to produce the best overall performance for each model, indicating that the probe can benefit from information encoded across various layers.

Once we compare the averaged results across syntactic representations, a preference for UD emerges, starting in layer 3 in BERT and layer 2 in ELMo. We observe the max difference in favor of UD in layer 7 for BERT, where the probe performs 3 UAS points better than SUD, and in the weighted average (layer 13), with 4 UAS points. The difference for the 13th BERT and 3rd ELMo layers is statistically significant at $p \leq 0.05$ (Wilcoxon signed ranks test). A further look at differences across languages reveals that, while most languages tend to overwhelmingly prefer UD, there are some that do not: Basque, Turkish, and, to a lesser extent, Finnish. Furthermore, the preference towards SUD in these languages tends to be most pronounced in the first four and last two layers of BERT. However, in the layers where we tend to observe the higher UAS

[7] It is important to note that layer 0 for ELMo is the non-recurrent embedding layer which contains no contextual information.

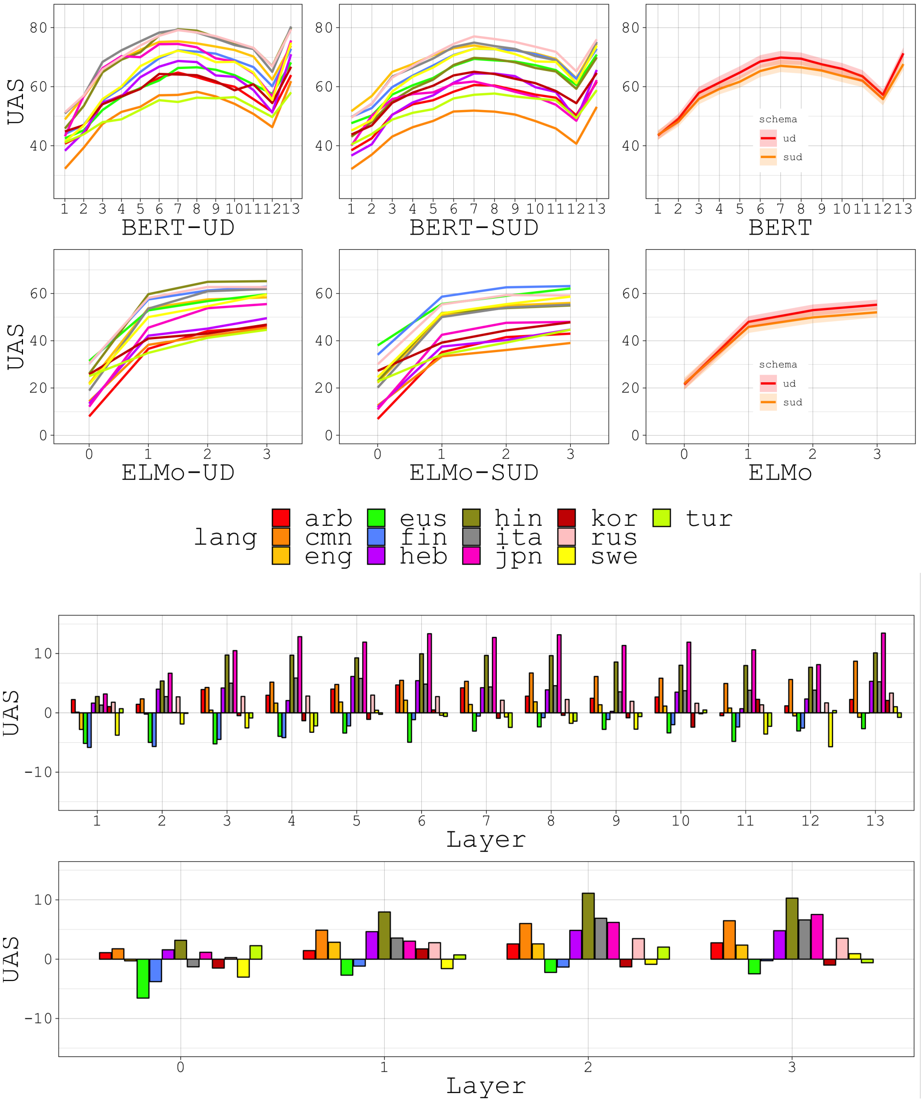


Figure VI.3: Probe results per model, layer, and language. First two rows depict UAS per layer and language for BERT and ELMo, with average performance and error over UD/SUD in 3rd column. Bottom two rows depict the difference in UAS across UD (+) and SUD (−) per model.

overall (7–8), this is minimized for Basque/Turkish and almost eliminated for Finnish. Indeed, we see the strongest preferences for UD in these layers overall, where Italian and Japanese are overwhelmingly pro-UD, to the order of 10+ UAS points.

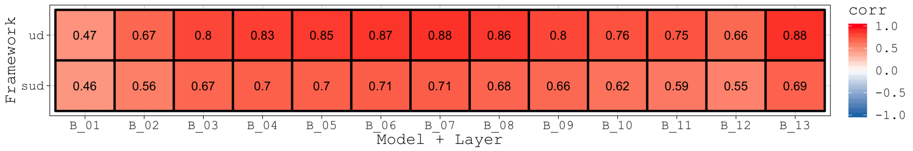


Figure VI.4: Pearson correlation between UD/SUD probing accuracy and supervised UAS, per layer.

VI.6.1 Controlling for Treebank Size

Overall, we note that some languages consistently achieve higher accuracy, like Russian with 71/69 UAS for UD/SUD for BERT, while others fare poorly, like Turkish (52/43) and Chinese (51/46). In the case of these languages, one can observe an obvious relation to the size of our reference treebanks, where Russian is by far the largest and Turkish and Chinese are the smallest. To test the extent to which training set size affects probing accuracy, we trained our probe on the same treebanks, truncated to the size of the smallest one — Turkish, with 3664 sentences. Though we did observe a decline in accuracy in the largest treebanks (e.g. Russian, Finnish, and English) for some layers, the difference in aggregate was minimal. Furthermore, the magnitude of the difference in UD and SUD probing accuracy was almost identical to that of the probes trained on full treebanks, speaking to the validity of our findings. We refer the reader to Appendix VI.A for these results.

VI.6.2 Connection to Supervised Parsing

Given that our findings seem to generally favor UD, another question we might ask is: are SUD treebanks simply harder to parse? This may seem like a straightforward hypothesis, given SUD's tendency to produce higher trees in aggregate, which may affect parsing accuracy — even in the fully supervised case. To test this, we trained UD and SUD parsers using the UDify model (Kondratyuk and Straka, 2019), which employs a biaffine attention decoder (Dozat and Manning, 2016) after fine-tuning BERT representations (similar to our 13th layer). The results showed a slightly higher average UAS for UD (89.9 vs. 89.6) and a slightly higher LAS for SUD (86.8 vs. 86.5). Neither difference is statistically significant (Wilcoxon signed ranks test), which seems to rule out an alternative explanation in terms of learnability. We include the full range of results in Appendix VI.B.

In addition to this, we tested how well each framework's probing accuracy related to supervised UAS across languages. We computed this measure by taking the Pearson correlation of each BERT probe's layer accuracy (per-language) with its respective framework accuracy. All correlations proved to be significant at $p \leq 0.05$, with the exception of UD and SUD at layer 1. Figure VI.4 displays these results. Here, we observe that probing accuracies correlate more strongly with supervised UAS for UD than for SUD. We can interpret this to mean that the rate at which trees are decoded by the UD probe is more indicative of how well they can be parsed given a full view of their structure, rather than vice-versa. Although correlation is an indirect measure here, we can still accept it to be in support of our general findings.

VI.6.3 Parts of Speech

In order to gain a better understanding of these probing patterns, we move on to an error analysis over the dev sets of each treebank, as fit by the averaged models. Figure VI.5 shows probe accuracy for different models (BERT/ELMo) and syntactic representations (UD/SUD) when attaching words of specific part-of-speech categories to their heads. The general pattern is that we observe higher accuracy for UD for both models on all categories, the only exceptions being a slightly higher accuracy for both models on PRON and for ELMo on VERB and X.[8] However, the differences are generally greater for function words, in particular ADP, AUX, SCONJ, PART and DET. In some respects, this is completely expected given the different treatment of these words in UD and SUD, and we can use the case of adpositions (ADP) to illustrate this. In UD, the preposition *from* in a phrase like *from the room* is simply attached to the noun *room*, which is in general a short relation that is easy to identify. In SUD, the relation between the preposition and the noun is reversed, and the preposition now has to be attached to whatever the entire phrase modifies, which often means that difficult attachment ambiguities need to be resolved. However, exactly the same ambiguities need to be resolved for nominal words (NOUN, PRON, PROPN) in the UD representation, but there is no corresponding drop in accuracy for these classes in UD (except very marginally for PRON). Similar remarks can be made for other function word categories, in particular AUX, SCONJ and PART. It thus seems that the UD strategy of always connecting content words directly to other content words, instead of sometimes having these relations mediated by function words, results in higher accuracy overall when applying the probe to the representations learned by BERT and ELMo.

The behavior of different part-of-speech classes can also explain some of the differences observed across languages. In particular, as can be seen in Table VI.1, most of the languages that show a clear preference for UD — Chinese, Hebrew, Hindi, Italian and Japanese — are all characterized by a high proportion of adpositions. Conversely, the three languages that exhibit the opposite trend — Basque, Finnish and Turkish — have a very low proportion of adpositions. The only language that does not fit this pattern is Chinese, which has a low percentage of adpositions but nevertheless shows a clear preference for UD. Finally, it is worth noting that Korean shows no clear preference for either representation despite having a very low proportion of adpositions (as well as other function words), but this is due to the more coarse-grained word segmentation of the Korean treebank, which partly incorporates function words into content word chunks.[9]

[8] The X category is unspecified and extremely rare.

[9] This is reflected also in the exceptionally high proportion of direct content word relations; cf. Table VI.1.

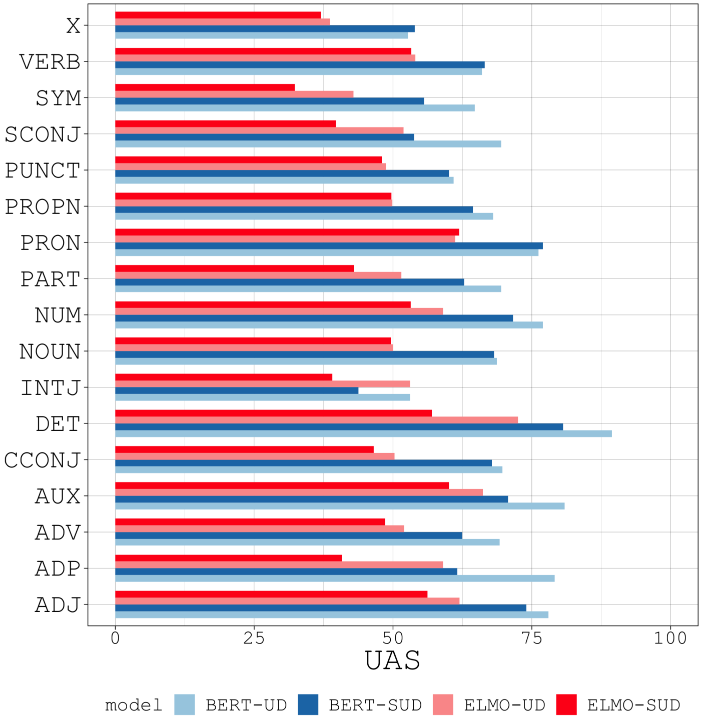


Figure VI.5: UAS accuracy for the average models (BERT 13, ELMo 3) on incoming dependencies of different part-of-speech categories.

VI.6.4 Sentence and Tree Properties

Figure VI.6 depicts probing accuracy across different sentence lengths, dependency lengths, and distances to root. It is apparent that, despite the absolute differences between models, the relative differences between representations are strikingly consistent in favor of UD. For example, while the probe shows identical accuracy for the two representations for sentences of length 1–10, SUD decays more rapidly with increasing sentence length. Furthermore, while the SUD probe is slightly more accurate at detecting sentence roots and their immediate dependencies, we observe a consistent advantage for dependencies of length 2+, until dropping off for the longest length bin of 10+. Though Table VI.1 indicates that UD dependencies are slightly longer than those of SUD, this factor does not appear to influence the probe, as there are no significant correlations between differences in average dependency length and differences in UAS.

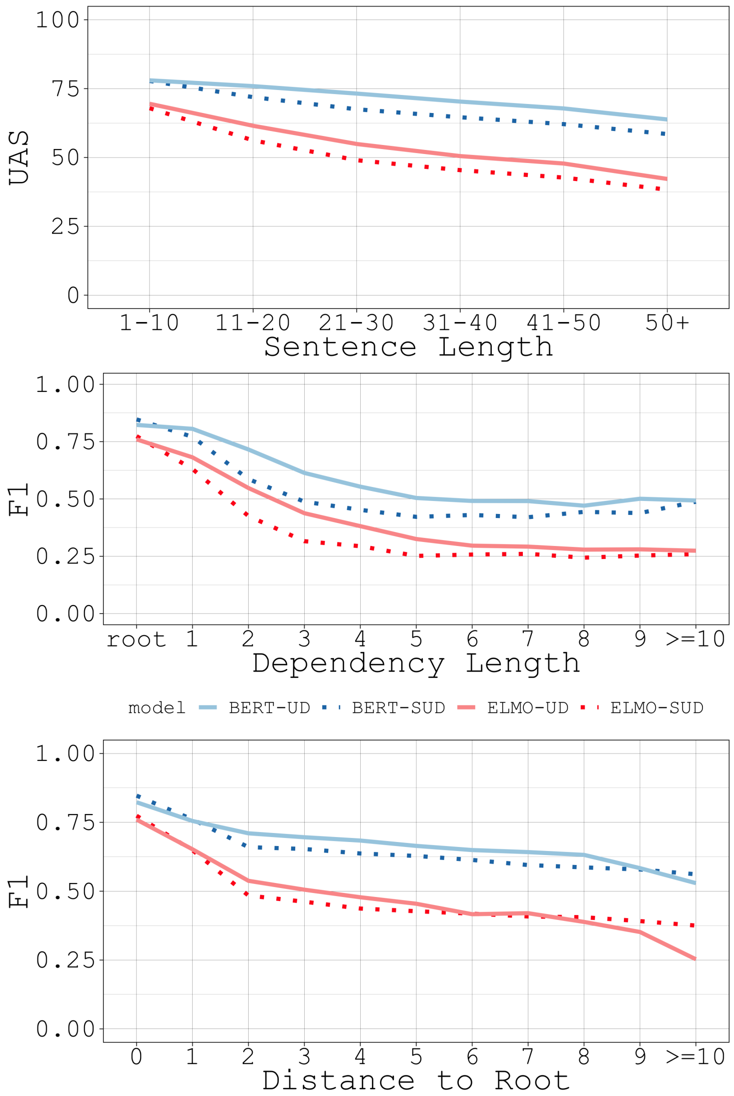


Figure VI.6: UAS across sentence length bins (top); F1 across varying dependency lengths (middle); F1 across varying distances to root (bottom)

We observe a similar curve for varying distances to root, where the SUD probe performs slightly better than UD at the shortest distance, but decays faster for nodes higher in the tree. In general, UD trees have lower height than SUD (see Table VI.1), which implies that tree height could be a major factor at play here. To verify this, we conducted a Pearson correlation test between the average increase in height from UD to SUD and the difference of the UD/SUD probe UAS per language. This test returned $\rho = 0.82, p < 0.001$, indicating that height is indeed crucial in accurately decoding trees across the two formalisms.

In an attempt to visualize how this may play out across languages, we plotted the per-sentence difference in probing accuracy between UD/SUD as a function of the difference in height of the respective gold UD/SUD trees. Figure VI.7 depicts these results for BERT, where the x-axis indicates how many nodes higher a SUD tree is with respect to its reference UD tree.

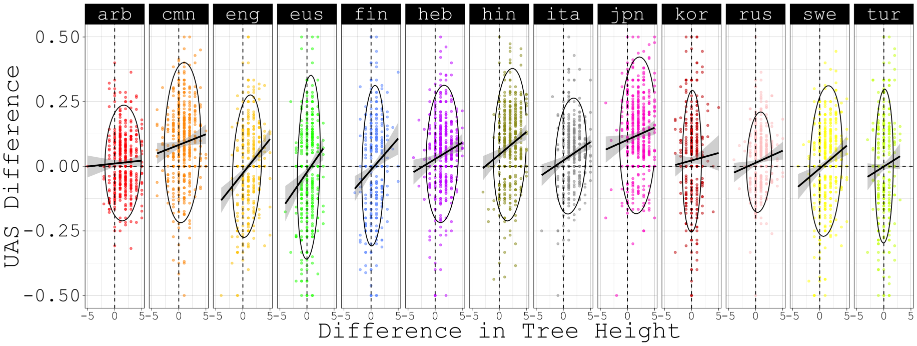


Figure VI.7: Differences in the BERT probe's UAS (UD +, SUD −) as a function of tree height per number of nodes (higher SUD tree +, higher UD tree −), with smoothed means and 95% confidence ellipses as implemented in `ggplot2`)

It is apparent from Figure VI.7 that the preference for UD can be largely explained via its lower tree height. If we first examine Korean, the segmentation of which results in the smallest difference in height overall, we observe a distribution that is roughly centered around zero on both axes. If we instead refer to the UD-preferring languages (Chinese, Hebrew, Hindi, Italian, and Japanese), we notice a strong skew of distributions towards the top right of the plot. This indicates (i) that the trees in these samples are higher for SUD and (ii) that the corresponding sentences are easier to decode in UD. By contrast, for the SUD-preferring languages (Basque, Finnish, and Turkish), we observe narrow distributions centered around 0 (similar to that of Korean), indicating minimal variation in tree height between UD and SUD. What these language have in common is an agglutinative morphology, which means that they rely more on morphological inflection to indicate relationships between content words, rather than separate function words. Sentences in these languages are therefore less susceptible to variations in tree height, by mere virtue of being shorter and possessing fewer relations that are likely be a better fit for UD, like those concerning adpositions. We speculate that it is this inherent property that explains the layerwise preference for SUD (though a general indifference in aggregate), allowing for some language-specific properties, like the crucial role of auxiliaries in Basque, to be easier to probe for in SUD. Conversely, with this in mind, it becomes easy to motivate the high preference for UD across some languages, given that they are not agglutinating and make heavy use of function words. If we take the probe to be a proper decoding of a model's representational space, the encoding of syntactic structure according to an

SUD-style analysis then becomes inherently more difficult, as the model is required to attend to hierarchy between words higher in the tree. Interestingly, however, this does not seem to correspond to an increased difficulty in the case of supervised parsing, as observed earlier.

VI.7 Conclusion and Future Work

We have investigated the extent to which the syntactic structure captured by neural language models aligns with different styles of analysis, using UD treebanks and their SUD conversions as proxies. We have extended the structural probe of Hewitt and Manning (2019) to extract directed, rooted trees and fit it on pretrained BERT and ELMo representations for 13 languages. Ultimately, we observed a better overall fit for the UD-style formalism across models, layers, and languages, with some notable exceptions. For example, while the Chinese, Hebrew, Hindi, Italian, and Japanese models proved to be overwhelmingly better-fit for UD, Basque aligned more with SUD, and Finnish, Korean and Turkish did not exhibit a clear preference. Furthermore, an error analysis revealed that, when attaching words of various part-of-speech tags to their heads, UD fared better across the vast majority of categories, most notably adpositions and determiners. Related to this, we found a strong correlation between differences in average tree height and the tendency to prefer one framework over the other. This suggested a tradeoff between morphological complexity — where differences in tree height between UD and SUD are minimal and probing accuracy similar — and a high proportion of function words — where SUD trees are significantly higher and probing accuracy favors UD.

For future work, besides seeking a deeper understanding of the interplay of linguistic factors and tree shape, we want to explore probes that combine the distance and depth assumptions into a single transformation, rather than learning separate probes and combining them post-hoc, as well as methods for alleviating treebank supervision altogether. Lastly, given recent criticisms of probing approaches in NLP, it will be vital to revisit the insights produced here within a non-probing framework, for example, using Representational Similarity Analysis (RSA) (Chrupała and Alishahi, 2019) over symbolic representations from treebanks and their encoded representations.

Acknowledgements

We want to thank Miryam De Lhoneux, Paola Merlo, Sara Stymne, and Dan Zeman and the ACL reviewers and area chairs for valuable feedback on preliminary versions of this paper. We acknowledge the computational resources provided by CSC in Helsinki and Sigma2 in Oslo through NeIC-NLPL (www.nlpl.eu). Joakim Nivre's contributions to this work were supported by grant 2016-01817 of the Swedish Research Council.

References

Abnar, S. et al. (2019). “Blackbox Meets Blackbox: Representational Similarity & Stability Analysis of Neural Language Models and Brains”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics.

Belinkov, Y. et al. (2017). “What do Neural Machine Translation Models Learn about Morphology?” en. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics.

Blevins, T. et al. (2018). “Deep RNNs Encode Soft Hierarchical Syntax”. en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Bresnan, J. (2000). *Lexical-Functional Syntax*. Blackwell.

Che, W. et al. (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*.

Chrupała, G. and Alishahi, A. (2019). “Correlating Neural and Symbolic Representations of Language”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Chu, Y. J. and Liu, T. H. (1965). “On the Shortest Arborescence of a Directed Graph”. In: *Science Sinica* vol. 14.

Conneau, A. et al. (2018). “What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties”. en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Devlin, J. et al. (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.

Dozat, T. and Manning, C. D. (2016). “Deep biaffine attention for neural dependency parsing”. In: *arXiv preprint arXiv:1611.01734*.

Ettinger, A. et al. (2016). “Probing for semantic evidence of composition by means of simple classification tasks”. In: *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics.

Gardner, M. et al. (2018). “AllenNLP: A Deep Semantic Natural Language Processing Platform”. In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*.

Gauthier, J. and Levy, R. (2019). “Linking artificial and human neural representations of language”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Nat-*

ural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics.

Gerdes, K. et al. (2018). "SUD or Surface-Syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD". In: *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*.

Hewitt, J. and Liang, P. (2019). "Designing and Interpreting Probes with Control 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.

Hewitt, J. and Manning, C. D. (2019). "A structural probe for finding syntax in word representations". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Hupkes, D. et al. (2018). "Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure". In: *Journal of Artificial Intelligence Research* vol. 61.

Kaplan, R. and Bresnan, J. (1982). "Lexical-Functional Grammar: A Formal System for Grammatical Representation". In: *The Mental Representation of Grammatical Relations*. Ed. by Bresnan, J. MIT Press.

Kondratyuk, D. and Straka, M. (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.

Kornblith, S. et al. (2019). "Similarity of neural network representations revisited". In.

Kulmizev, A. et al. (2019). "Deep Contextualized Word Embeddings in Transition-Based and Graph-Based Dependency Parsing - A Tale of Two Parsers Revisited". 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.

McDonald, R. et al. (2005). "Non-Projective Dependency Parsing using Spanning Tree Algorithms". In: *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

Mel'čuk, I. (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press.

Nivre, J. et al. (2016). "Universal Dependencies v1: A Multilingual Treebank Collection". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA).

Nivre, J. et al. (2019). *Universal Dependencies 2.4*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Peters, M. et al. (2018a). “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.

Peters, M. et al. (2018b). “Dissecting Contextual Word Embeddings: Architecture and Representation”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Prasad, G. et al. (2019). “Using Priming to Uncover the Organization of Syntactic Representations in Neural Language Models”. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics.

Raghu, M. et al. (2017). “SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability”. In: *Advances in Neural Information Processing Systems*.

Saphra, N. and Lopez, A. (2018). “Understanding learning dynamics of language models with SVCCA”. In: *arXiv preprint arXiv:1811.00225*.

Tenney, I. et al. (2019a). “BERT rediscovers the classical NLP pipeline”. In: *arXiv preprint arXiv:1905.05950*.

Tenney, I. et al. (2019b). “What do you learn from context? Probing for sentence structure in contextualized word representations”. In: *International Conference on Learning Representations*.

Appendix VI.A Controlling for Treebank Size

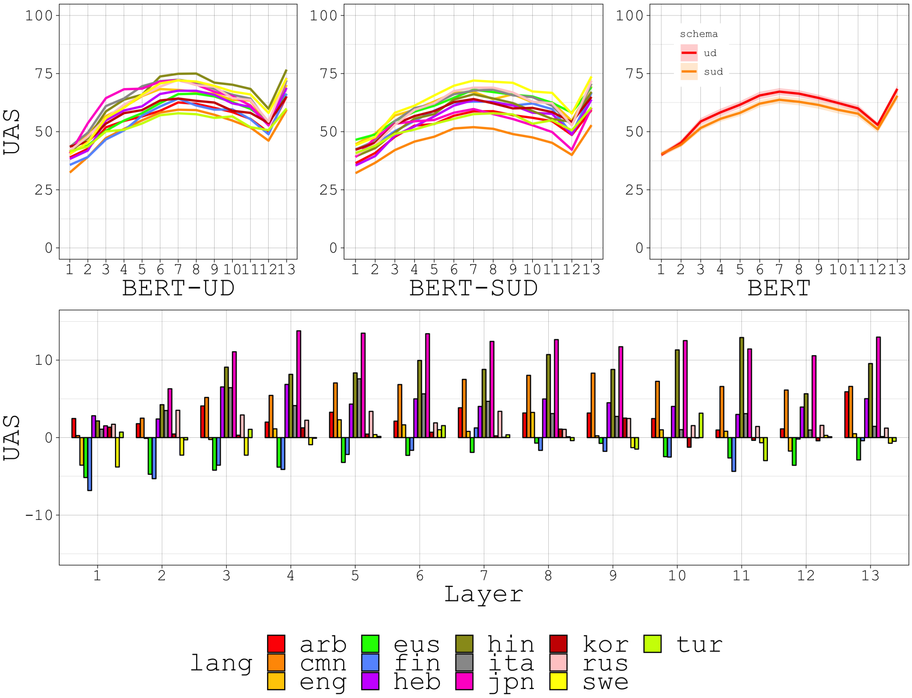


Figure VI.8: Probe results per framework, layer, and language, when trained on 3664 sentences. First row depicts UAS per layer and language for BERT, with average performance and error over UD/SUD in 3rd column. Bottom two row depicts the difference in UAS across UD (+) and SUD (−).

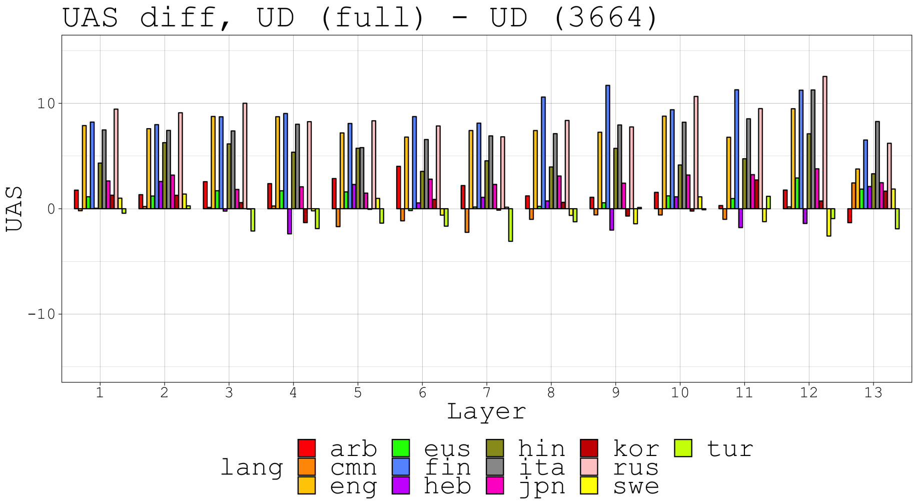


Figure VI.9: Difference in UAS across the UD probes trained on full data (+) and 3664 sentences (−).

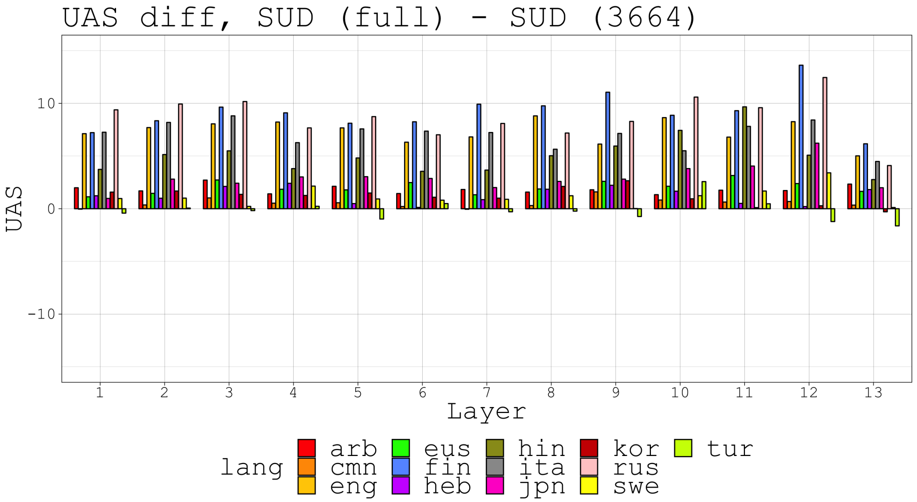


Figure VI.10: Difference in UAS across the SUD probes trained on full data (+) and 3664 sentences (−).

Appendix VI.B Connection to Supervised Parsing

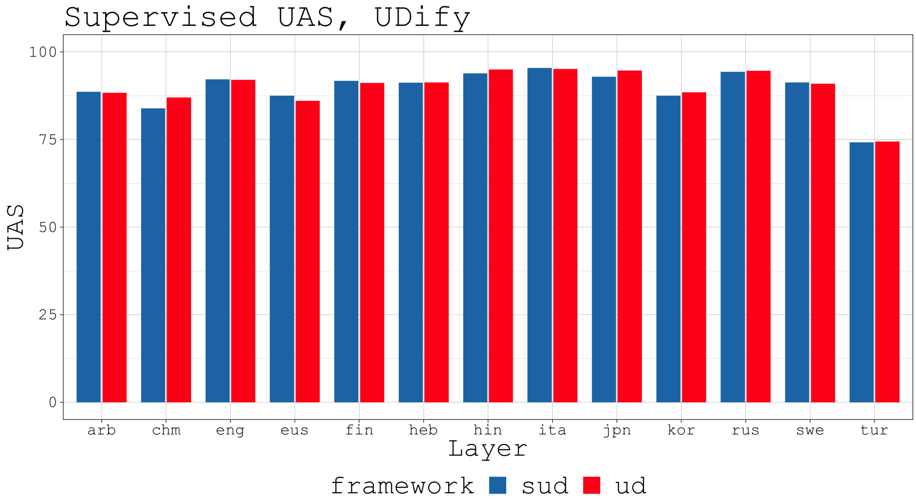


Figure VI.11: Supervised UDify UAS, UD and SUD, for all languages.

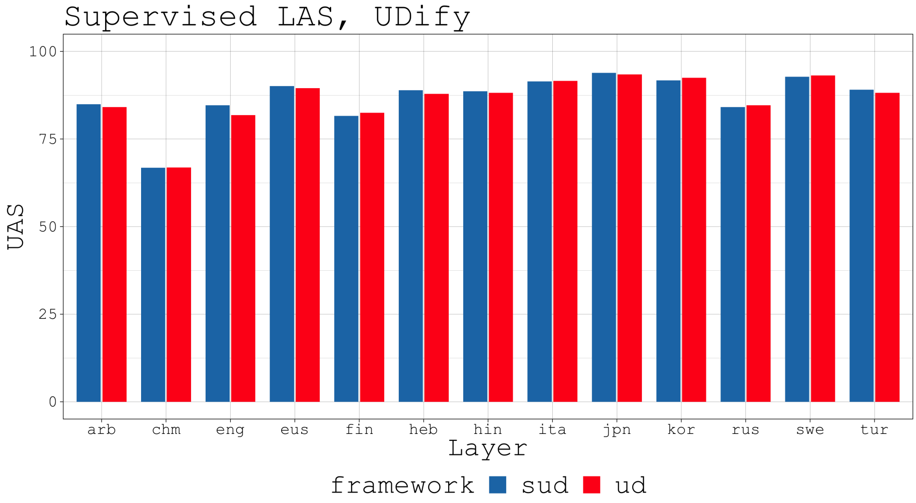


Figure VI.12: Supervised Udify LAS, UD and SUD, for all languages.

Paper VII

Attention Can Reflect Syntactic Structure If You Let It

***Vinit Ravishankar, *Artur Kulmizev, Mostafa Abdou, Anders Søgaard, Joakim Nivre**

Appears in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, April 2021, pp. 3031–3045. DOI: 10.18653/v1/2021.eacl-main.264.

Abstract

Since the popularization of the Transformer as a general-purpose feature encoder for NLP, many studies have attempted to decode linguistic structure from its novel multi-head attention mechanism. However, much of such work focused almost exclusively on English — a language with rigid word order and a lack of inflectional morphology. In this study, we present decoding experiments for multilingual BERT across 18 languages in order to test the generalizability of the claim that dependency syntax is reflected in attention patterns. We show that full trees can be decoded above baseline accuracy from single attention heads, and that individual relations are often tracked by the same heads across languages. Furthermore, in an attempt to address recent debates about the status of attention as an explanatory mechanism, we experiment with fine-tuning mBERT on a supervised parsing objective while freezing different series of parameters. Interestingly, in steering the objective to learn explicit linguistic structure, we find much of the same structure represented in the resulting attention patterns, with interesting differences with respect to which parameters are frozen.

Contents

VII.1 Introduction

In recent years, the attention mechanism proposed by Bahdanau et al. (2015) has become an indispensable component of many NLP systems. Its widespread adoption was, in part, heralded by the introduction of the Transformer architecture (Vaswani et al., 2017b), which constrains a soft alignment to be learned across discrete states in the input (self-attention), rather than across input and output (e.g., Rocktäschel et al., 2015; Xu et al., 2015). The Transformer has, by now, supplanted the popular LSTM (Hochreiter and Schmidhuber, 1997) as NLP's feature-encoder-of-choice, largely due to its compatibility with parallelized training regimes and ability to handle long-distance dependencies.

Certainly, the nature of attention as a distribution over tokens lends itself to a straightforward interpretation of a model's inner workings. Bahdanau et al. (2015) illustrate this nicely in the context of `seq2seq` machine translation, showing that the attention learned by their models reflects expected cross-lingual idiosyncrasies between English and French, e.g., concerning word order. With self-attentive Transformers, interpretation becomes slightly more difficult, as attention is distributed across words within the input itself. This is further compounded by the use of multiple layers and heads, each combination of which yields its own alignment, representing a different (possibly redundant) view of the data. Given the similarity of such attention matrices to the score matrices employed in arc-factored dependency parsing (McDonald et al., 2005a; McDonald et al., 2005b), a salient question concerning interpretability becomes: Can we expect some combination of these parameters to capture linguistic structure in the form of a dependency tree, especially if the model performs well on NLP tasks? If not, can we relax the expectation and examine the extent to which subcomponents of the linguistic structure, such as subject-verb relations, are represented? This prospect was first posed by Raganato, Tiedemann, et al. (2018) for MT encoders, and later explored by Clark et al. (2019) for BERT. Ultimately, the consensus of these and other studies (Htut et al., 2019; Limisiewicz et al., 2020; Voita et al., 2019) was that, while there appears to exist no "generalist" head responsible for extracting full dependency structures, standalone heads often specialize in capturing individual grammatical relations.

Unfortunately, most of such studies focused their experiments entirely on English, which is typologically favored to succeed in such scenarios due to its rigid word order and lack of inflectional morphology. It remains to be seen whether the attention patterns of such models can capture structural features across typologically diverse languages, or if the reported experiments on English are a misrepresentation of local positional heuristics as such. Furthermore, though previous work has investigated how attention patterns might change after fine-tuning on different tasks (Htut et al., 2019), a recent debate about attention as an explanatory mechanism (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019) has cast the entire enterprise in doubt. Indeed, it remains to be seen whether fine-tuning on an explicit structured prediction task, e.g. dependency parsing, can force attention to represent the structure being learned, or if the patterns observed in pretrained models are not altered in any meaningful way.

To address these issues, we investigate the prospect of extracting linguistic structure from the attention weights of multilingual Transformer-based language models. In light of the surveyed literature, our research questions are as follows:

1. Can we decode dependency trees for some languages better than others?
2. Do the same layer–head combinations track the same relations across languages?
3. How do attention patterns change after fine-tuning with explicit syntactic annotation?
4. Which components of the model are involved in these changes?

In answering these questions, we believe we can shed further light on the (cross-)linguistic properties of Transformer-based language models, as well as address the question of attention patterns being a reliable representation of linguistic structure.

VII.2 Attention as Structure

Transformers The focus of the present study is mBERT, a multilingual variant of the exceedingly popular language model (Devlin et al., 2019). BERT is built upon the Transformer architecture (Vaswani et al., 2017a), which is a self-attention-based encoder-decoder model (though only the encoder is relevant to our purposes). A Transformer takes a sequence of vectors $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n}]$ as input and applies a positional encoding to them, in order to retain the order of words in a s entence. These inputs are then transformed into query (Q), key (K), and value (V) vectors via three separate linear transformations and passed to an attention mechanism. A single attention head computes scaled dot-product attention between K and Q, outputting a weighted sum of V:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{VII.1}$$

For multihead attention (MHA), the same process is repeated for k heads, allowing the model to jointly attend to information from different representation subspaces at different positions (Vaswani et al., 2017a). Ultimately, the output of all heads is concatenated and passed through a linear projection W^O:

$$H_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{VII.2}$$

$$\text{MHA}(Q, K, V) = \text{concat}(H_1, H_2, ..., H_k)W^O \tag{VII.3}$$

Every layer also consists of a feed-forward network (FFN), consisting of two Dense layers with ReLU activation functions. For each layer, therefore, the output of MHA is passed through a LayerNorm with residual connections, passed through FFN, and then through another LayerNorm with residual connections.

Searching for structure Often, the line of inquiry regarding interpretability in NLP has been concerned with extracting and analyzing linguistic information from neural network models of language (Belinkov and Glass, 2019). Recently, such investigations have targeted Transformer models Hewitt and Manning, 2019; Rosa and Mareček, 2019; Tenney et al., 2019, at least in part because the self-attention mechanism employed by these models offers a possible window into their inner workings. With large-scale machine translation and language models being openly distributed for experimentation, several researchers have wondered if self-attention is capable of representing syntactic structure, despite not being trained with any overt parsing objective.

In pursuit of this question, Raganato, Tiedemann, et al. (2018) applied a maximum-spanning-tree algorithm over the attention weights of several trained MT models, comparing them with gold trees from Universal Dependencies (Nivre et al., 2016; Nivre et al., 2020). They found that, while the accuracy was not comparable to that of a supervised parser, it was nonetheless higher than several strong baselines, implying that some structure was consistently represented. Clark et al. (2019) corroborated the same findings for BERT when decoding full trees, but observed that individual dependency relations were often tracked by specialized heads and were decodable with much higher accuracy than some fixed-offset baselines. Concurrently, Voita et al. (2019) made a similar observation about heads specializing in specific dependency relations, proposing a coarse taxonomy of head attention functions: *positional*, where heads attend to adjacent tokens; *syntactic*, where heads attend to specific syntactic relations; and *rare words*, where heads point to the least frequent tokens in the sentence. Htut et al. (2019) followed Raganato, Tiedemann, et al. (2018) in decoding dependency trees from BERT-based models, finding that fine-tuning on two classification tasks did not produce syntactically plausible attention patterns. Lastly, Limisiewicz et al. (2020) modified UD annotation to better represent attention patterns and introduced a supervised head-ensembling method for consolidating shared syntactic information across heads.

Does attention have explanatory value? Though many studies have yielded insight about how attention behaves in a variety of models, the question of whether it can be seen as a "faithful" explanation of model predictions has been subject to much recent debate. For example, Jain and Wallace (2019) present compelling arguments that attention does not offer a faithful explanation of predictions. Primarily, they demonstrate that there is little correlation between standard feature importance measures and attention weights. Furthermore, they contend that there exist *counterfactual* attention distributions, which are substantially different from learned attention weights but that do not alter a model's predictions. Using a similar methodology, Serrano and Smith (2019) corroborate that attention does not provide an adequate account of an input component's importance.

In response to these findings, Wiegreffe and Pinter (2019) question the assumptions underlying such claims. Attention, they argue, is not a *primitive*, i.e., it cannot be detached from the rest of a model's components as is done in the experiments of Jain and Wallace (2019). They propose a set of four analyses to test whether a given model's attention mechanism can provide meaningful explanation and demonstrate that the alternative attention distributions found via adversarial training methods do, in fact, perform poorly compared to standard attention mechanisms. On a theoretical level, they argue that, although attention weights do not give an *exclusive* "faithful" explanation, they do provide a meaningful *plausible* explanation.

This discussion is relevant to our study because it remains unclear whether or not attending to syntactic structure serves, in practice, as plausible explanation for model behavior, or whether or not it is even capable of serving as such. Indeed, the studies of Raganato, Tiedemann, et al. (2018) and Clark et al. (2019) relate a convincing but incomplete picture — tree decoding accuracy just marginally exceeds baselines and various relations tend to be tracked across varying heads and layers. Thus, our fine-tuning experiments (detailed in the following section) serve to enable an "easy" setting wherein we explicitly inform our models of the same structure that we are trying to extract. We posit that, if, after fine-tuning, syntactic structures were still *not* decodable from the attention weights, one could safely conclude that these structures are being stored via a non-transparent mechanism that may not even involve attention weights. Such an insight would allow us to conclude that attention weights cannot provide even a plausible explanation for models relying on syntax.

VII.3 Experimental Design

To examine the extent to which we can decode dependency trees from attention patterns, we run a tree decoding algorithm over mBERT's attention heads — before and after fine-tuning via a parsing objective. We surmise that doing so will enable us to determine if attention can be interpreted as a reliable mechanism for capturing linguistic structure.

VII.3.1 Model

We employ mBERT[1] in our experiments, which has been shown to perform well across a variety of NLP tasks (Hu et al., 2020; Kondratyuk and Straka, 2019a) and capture aspects of syntactic structure cross-lingually (Chi et al., 2020; Pires et al., 2019). mBERT features 12 layers with 768 hidden units and 12 attention heads, with a joint WordPiece sub-word vocabulary across languages. The model was trained on the concatenation of WikiDumps for the top 104 languages with the largest Wikipedias,where principled sampling was employed to enforce a balance between high- and low-resource languages.

[1] https://github.com/google-research/bert

VII.3.2 Decoding Algorithm

For decoding dependency trees, we follow Raganato, Tiedemann, et al. (2018) in applying the Chu-Liu-Edmonds maximum spanning tree algorithm (Chu, 1965) to every layer/head combination available in mBERT ($12 \times 12 = 144$ in total). In order for the matrices to correspond to gold treebank tokenization, we remove the cells corresponding to the BERT delimiter tokens (`[CLS]` and `[SEP]`). In addition to this, we sum the columns and average the rows corresponding to the constituent subwords of gold tokens, respectively (Clark et al., 2019). Lastly, since attention patterns across heads may differ in whether they represent heads attending to their dependents or vice versa, we take our input to be the element-wise product of a given attention matrix and its transpose ($A \circ A^\top$). We liken this to the joint probability of a head attending to its dependent and a dependent attending to its head, similarly to Limisiewicz et al. (2020). Per this point, we also follow Htut et al. (2019) in evaluating the decoded trees via Undirected Unlabeled Attachment Score (UUAS) — the percentage of undirected edges recovered correctly. Since we discount directionality, this is effectively a less strict measure than UAS, but one that has a long tradition in unsupervised dependency parsing since Klein and Manning (2004).

VII.3.3 Data

For our data, we employ the Parallel Universal Dependencies (PUD) treebanks, as collected in UD v2.4 (Nivre et al., 2019). PUD was first released as part of the CONLL 2017 shared task (Zeman et al., 2018), containing 1000 parallel sentences, which were (professionally) translated from English, German, French, Italian, and Spanish to 14 other languages. The sentences are taken from two domains, **news** and **wikipedia**, the latter implying some overlap with mBERT's training data (though we did not investigate this). We include all PUD treebanks except Thai.[2]

VII.3.4 Fine-Tuning Details

In addition to exploring pretrained mBERT's attention weights, we are also interested in how attention might be guided by a training objective that learns the exact tree structure we aim to decode. To this end, we employ the graph-based decoding algorithm of the biaffine parser introduced by Dozat and Manning (2016). We replace the standard BiLSTM encoder for this parser with the entire mBERT network, which we fine-tune with the parsing loss. The full parser decoder consists of four dense layers, two for head/child representations for dependency arcs (dim. 500) and two for head/child representations for dependency labels (dim. 100). These are transformed into the label space via a bilinear transform.

[2] Thai is the only treebank that does not have a non-PUD treebank available in UD, which we need for our fine-tuning experiments.

	AR	CS	DE	EN	ES	FI	FR	HI	ID	IT	JA	KO	PL	PT	RU	SV	TR	ZH
BASELINE	50	40	36	36	40	42	40	46	47	40	43	55	45	41	42	39	52	41
PRE	53 7-6	53 10-8	49 10-8	47 10-8	50 9-5	48 10-8	41 2-3	48 2-3	50 9-5	41 6-4	**45** 2-3	64 9-2	52 10-8	50 9-5	51 10-8	51 10-8	55 3-8	42 2-3
NONE	**76** 11-10	**78** 11-10	**76** 11-10	**71** 10-11	**77** 10-11	**66** 10-11	**45** 11-10	**72** 11-10	**75** 11-10	**58** 11-10	42 11-10	64 11-10	**75** 11-10	**76** 11-10	**75** 10-8	**74** 10-8	55 3-8	38 2-3
KEY	62 10-8	64 10-8	58 11-12	53 10-8	59 11-12	56 10-8	41 7-12	54 10-8	59 10-8	47 9-2	44 2-3	62 10-8	64 10-8	58 11-12	61 10-8	59 12-10	55 3-12	41 2-3
QUERY	69 11-4	74 10-8	70 11-4	66 11-4	73 11-4	63 10-8	42 11-4	62 11-4	67 11-4	54 11-4	**45** 2-3	65 10-8	72 11-4	70 11-4	70 10-8	68 11-4	56 10-8	42 2-3
KQ	71 11-4	76 11-4	70 11-4	65 11-4	74 11-4	62 11-4	43 10-11	64 11-4	69 11-4	55 11-4	44 2-3	64 11-4	73 11-4	73 11-4	69 11-4	69 11-4	55 11-4	41 2-3
VALUE	75 12-5	72 12-5	72 12-5	64 12-5	76 12-5	59 12-5	**45** 12-5	63 12-5	73 12-5	55 12-5	**45** 2-3	**66** 10-8	73 12-5	74 12-5	69 12-5	65 12-5	**57** 12-5	**42** 3-8
DENSE	68 11-10	71 11-10	65 11-10	60 10-8	67 12-10	61 11-10	42 10-8	65 11-10	66 11-10	49 9-5	44 3-12	64 11-10	70 11-10	64 12-5	67 11-10	64 11-10	55 11-10	40 3-12

Table VII.1: Adjacent-branching baseline and maximum UUAS decoding accuracy per PUD treebank, expressed as best score and best layer/head combination for UUAS decoding. PRE refers to basic mBERT model before fine-tuning, while all cells below correspond different fine-tuned models described in Section 3.4. Best score indicated in **bold**.

After training the parser, we can decode the fine-tuned mBERT parameters in the same fashion as described in Section VII.3.2. We surmise that, if attention heads are capable of tracking hierarchical relations between words in any capacity, it is precisely in this setting that this ability would be attested. In addition to this, we are interested in what individual *components* of the mBERT network are capable of steering attention patterns towards syntactic structure. We believe that addressing this question will help us not only in interpreting decisions made by BERT-based neural parsers, but also in aiding us developing syntax-aware models in general (Strubell et al., 2018; Swayamdipta et al., 2018). As such — beyond fine-tuning all parameters of the mBERT network (our basic setting) — we perform a series of ablation experiments wherein we update only one set of parameters per training cycle, e.g. the Query weights W_i^Q, and leave everything else frozen. This gives us a set of 6 models, which are described below. For each model, all non-BERT parser components are always left unfrozen.

- KEY: only the K components of the transformer are unfrozen; these are the representations of tokens that are paying attention *to* other tokens.
- QUERY: only the Q components are unfrozen; these, conversely, are the representations of tokens being paid attention to.
- KQ: both keys and queries are unfrozen.
- VALUE: semantic value vectors per token (V) are unfrozen; they are composed after being weighted with attention scores obtained from the K/Q matrices.
- DENSE: the dense feed-forward networks in the attention mechanism; all three per layer are unfrozen.
- NONE: The basic setting with nothing frozen; all parameters are updated with the parsing loss.

We fine-tune each of these models on a concatentation of all PUD treebanks for 20 epochs, which effectively makes our model multilingual. We do so in order to 1) control for domain and annotation confounds, since all PUD sentences are parallel and are natively annotated (unlike converted UD treebanks, for instance); 2) increase the number of training samples for fine-tuning, as each PUD treebank features only 1000 sentences; and 3) induce a better parser through multilinguality, as in Kondratyuk and Straka (2019b). Furthermore, in order to gauge the overall performance of our parser across all ablated settings, we evaluate on the test set of the largest non-PUD treebank available for each language, since PUD only features test partitions. When training, we employ a combined dense/sparse Adam optimiser, at a learning rate of $3 * 10^{-5}$. We rescale gradients to have a maximum norm of 5.

VII.4 Decoding mBERT Attention

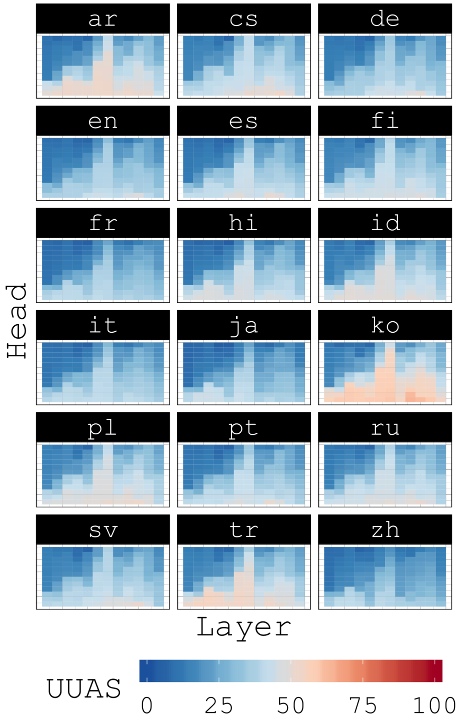
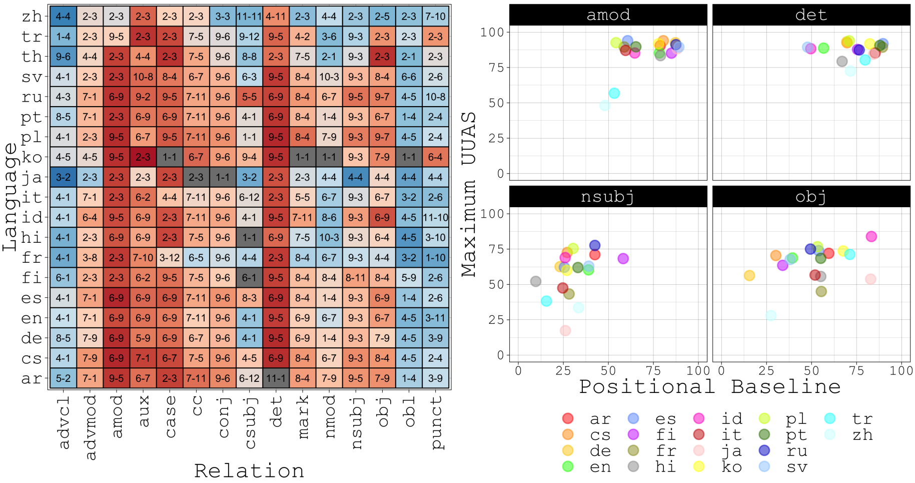


Figure VII.1: UUAS of MST decoding per layer and head, across languages. Heads (y-axis) are sorted by accuracy for easier visualization.

The second row of Table VII.1 (PRE) depicts the UUAS after running our decoding algorithm over mBERT attention matrices, per language. We see a familiar pattern to that in Clark et al. (2019) among others — namely that



	advcl	advmod	amod	aux	case	cc	conj	csubj	det	mark	nmod	nsubj	obj	obl	punct
zh	4-4	2-3	2-3	2-3	2-3	2-3	3-3	11-11	4-11	2-3	4-4	2-3	2-5	2-3	7-10
tr	1-4	2-3	9-5	2-3	2-3	7-5	9-6	9-12	9-5	4-2	3-6	9-3	2-3	2-3	2-3
th	9-6	4-4	2-3	4-4	2-3	7-5	9-6	8-8	2-3	7-5	2-1	9-3	2-3	2-1	2-3
sv	4-1	2-3	2-3	10-8	8-4	6-7	9-6	6-3	9-5	8-4	10-3	9-3	8-4	6-6	2-6
ru	4-3	7-1	6-9	9-2	9-5	7-11	9-6	5-5	6-9	8-4	6-7	9-5	9-7	4-5	10-8
pt	8-5	7-1	2-3	6-9	6-9	7-11	9-6	4-1	6-9	8-4	1-4	9-3	6-7	1-4	2-4
pl	4-1	2-3	9-5	6-7	9-5	7-11	9-6	1-1	9-5	8-4	7-9	9-3	9-7	4-5	2-4
ko	4-5	4-5	9-5	2-3	1-1	6-7	9-6	9-4	9-5	1-1	1-1	9-3	7-9	1-1	6-4
ja	3-2	2-3	2-3	2-3	2-3	2-3	1-1	3-2	2-3	2-3	4-4	4-4	4-4	4-4	4-4
it	4-1	7-1	2-3	6-2	4-4	7-11	9-6	6-12	2-3	5-5	6-7	9-3	6-7	3-2	2-6
id	4-1	6-4	9-5	6-9	2-3	7-11	9-6	4-1	9-5	7-11	8-6	9-3	6-9	4-5	11-10
hi	4-1	2-3	6-9	6-9	2-3	7-5	9-6	1-1	6-9	7-5	10-3	9-3	6-4	4-5	3-10
fr	4-1	3-8	2-3	7-10	3-12	6-5	9-6	4-4	2-3	8-4	6-7	9-3	4-4	3-2	1-10
fi	6-1	2-3	2-3	6-2	9-5	7-5	9-6	6-1	9-5	8-4	8-4	8-11	8-4	5-9	2-6
es	4-1	7-1	6-9	6-9	6-9	7-11	9-6	8-3	6-9	8-4	1-4	9-3	6-9	1-4	2-6
en	4-1	7-1	6-9	6-9	6-9	7-11	9-6	4-1	9-5	6-9	6-7	9-3	6-7	4-5	3-11
de	8-5	7-9	6-9	5-9	6-9	6-7	9-6	4-1	9-5	6-9	1-4	9-3	7-9	4-5	3-9
cs	4-1	7-9	6-9	7-1	6-7	7-5	9-6	4-5	6-9	8-4	6-7	9-3	8-4	4-5	2-4
ar	5-2	7-1	9-5	6-7	2-3	7-11	9-6	6-12	11-1	8-4	7-9	9-5	7-9	1-4	3-9

Figure VII.2: Left: UUAS per relation across languages (best layer/head combination indicated in cell). Right: Best UUAS as a function of best positional baseline (derived from the treebank), selected relations.

attention patterns extracted directly from mBERT appear to be incapable of decoding dependency trees beyond a threshold of 50–60% UUAS accuracy. However, we also note that, in all languages, the attention-decoding algorithm outperforms a BASELINE (row 1) that draws an (undirected) edge between any two adjacent words in linear order, which implies that some non-linear structures are captured with regularity. Indeed, head 8 in layer 10 appears to be particularly strong in this regard, returning the highest UUAS for 7 languages. Interestingly, the accuracy patterns across layers depicted in Figure VII.1 tend to follow an identical trend for all languages, with nearly all heads in layer 7 returning high within-language accuracies.

It appears that attention for some languages (Arabic, Czech, Korean, Turkish) is comparatively easier to decode than others (French, Italian, Japanese, Chinese). A possible explanation for this result is that dependency relations between content words, which are favored by the UD annotation, are more likely to be adjacent in the morphologically rich languages of the first group (without intervening function words). This assumption seems to be corroborated by the high baseline scores for Arabic, Korean and Turkish (but not Czech). Conversely, the low baselines scores and the likewise low decoding accuracies for the latter four languages are difficult to characterize. Indeed, we could not identify what factors — typological, annotation, tokenization or otherwise — would set French and Italian apart from the remaining languages in terms of score. However, we hypothesize that the tokenization and our treatment of subword tokens plays a part in attempting to decode attention from Chinese and Japanese representations. Per the mBERT documentation,[3] Chinese and Japanese Kanji

[3] https://github.com/google-research/bert/blob/master/multilingual.md

character spans within the CJK Unicode range are character-tokenized. This lies in contrast with all other languages (Korean Hangul and Japanese Hiragana and Katakana included), which rely on whitespace and WordPiece (Wu et al., 2016). It is thus possible that the attention distributions for these two languages (at least where CJK characters are relevant) are devoted to composing words, rather than structural relations, which will distort the attention matrices that we compute to correspond with gold tokenization (e.g. by maxing rows and averaging columns).

Relation analysis We can disambiguate what sort of structures are captured with regularity by looking at the UUAS returned per dependency relation. Figure VII.2 (left) shows that adjectival modifiers (`amod`, mean UUAS = 85 ± 12) and determiners (`det`, 88 ± 6) are among the easiest relations to decode across languages. Indeed, words that are connected by these relations are often adjacent to each other and may be simple to decode if a head is primarily concerned with tracking linear order. To verify the extent to which this might be happening, we plot the aforementioned decoding accuracy as a function of select relations' positional baselines in Figure VII.2 (right). The positional baselines, in this case, are calculated by picking the most frequent offset at which a dependent occurs with respect to its head, e.g., -1 for `det` in English, meaning one position to the left of the head. Interestingly, while we observe significant variation across the positional baselines for `amod` and `det`, the decoding accuracy remains quite high.

In slight contrast to this, the core subject (`nsubj`, 58 ± 16 SD) and object (`obj`, 64 ± 13) relations prove to be more difficult to decode. Unlike the aforementioned relations, `nsubj` and `obj` are much more sensitive to the word order properties of the language at hand. For example, while a language like English, with Subject-Verb-Object (SVO) order, might have the subject frequently appear to the left of the verb, an SOV language like Hindi might have it several positions further away, with an object and its potential modifiers intervening. Indeed, the best positional baseline for English `nsubj` is 39 UUAS, while it is only 10 for Hindi. Despite this variation, the relation seems to be tracked with some regularity by the same head (layer 3, head 9), returning 60 UUAS for English and 52 for Hindi. The same can largely be said for `obj`, where the positional baselines return 51 ± 18. In this latter case, however, the heads tend to be much differently distributed across languages. Finally, he results for the `obj` relation provides some support for our earlier explanation concerning morphologically rich languages, as Arabic, Czech, Korean and Turkish all have among the highest accuracies (as well as positional baselines).

VII.5 Fine-Tuning Experiments

Next, we investigate the effect fine-tuning has on UUAS decoding. Row 3 in Table VII.1 (None) indicates that fine-tuning does result in large improvements to UUAS decoding across most languages, often by margins as high as $\sim 30\%$.

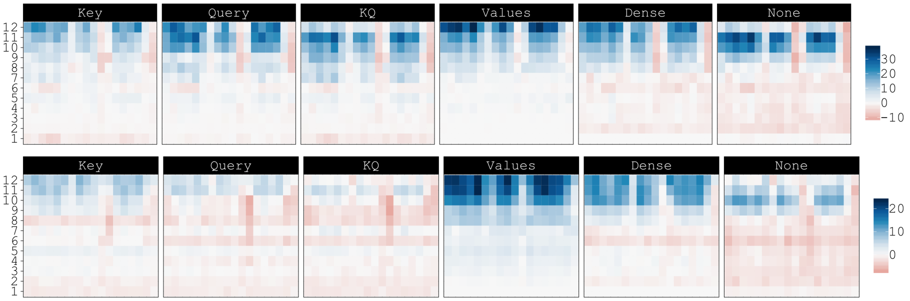


Figure VII.3: (Top) best scores across all heads, per language; (bottom) mean scores across all heads, per language. The languages (hidden from the X-axis for brevity) are, in order, *ar, cs, de, en, es, fi, fr, hi, id, it, ja, ko, pl, pt, ru, sv, tr, zh*

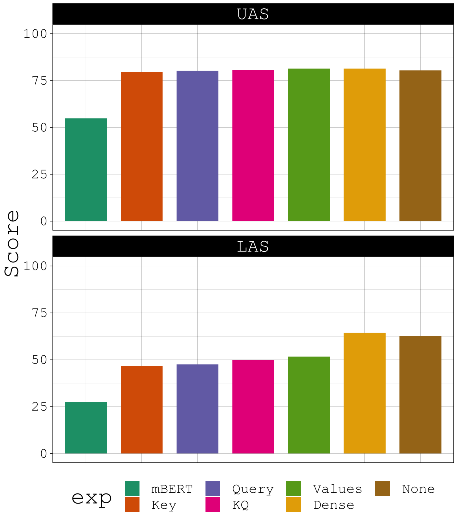


Figure VII.4: Mean UAS and LAS when evaluating different models on language-specific treebanks (Korean excluded due to annotation differences). mBERT refers to models where the entire mBERT network is frozen as input to the parser.

This shows that with an explicit parsing objective, attention heads are capable of serving as explanatory mechanisms for syntax; syntactic structure can be made to be transparently stored in the heads, in a manner that does not require additional probe fitting or parameterized transformation to extract.

Given that we do manage to decode reasonable syntactic trees, we can then refine our question — what components are capable of learning these trees? One obvious candidate is the key/query component pair, given that attention weights are a scaled softmax of a composition of the two. Figure VII.3 (top) shows the difference between pretrained UUAS and fine-tuned UUAS per layer, across models and languages. Interestingly, the best parsing accuracies do not appear to vary much depending on what component is frozen. We do see a clear trend, however, in that decoding the attention patterns of the fine-tuned model typically yields better UUAS than the pretrained model, particularly in the highest layers. Indeed, the lowest layer at which fine-tuning appears to improve decoding is layer 7. This implies that, regardless of which component remains frozen, the parameters facing any sort of significant and positive update tend to be those appearing towards the higher-end of the network, closer to the output.

For the frozen components, the best improvements in UUAS are seen at the final layer in VALUE, which is also the only model that shows consistent improvement, as well as the highest average improvement in mean scores[4] for the last few layers. Perhaps most interestingly, the mean UUAS (Figure VII.3 (bottom)) for our "attentive" components – keys, queries, and their combination – does not appear to have improved by much after fine-tuning. In contrast, the maximum does show considerable improvement; this seems to imply that although all components appear to be more or less equally capable of learning decodable heads, the attentive components, when fine-tuned, appear to sharpen fewer heads.

Note that the only difference between keys and queries in an attention mechanism is that keys are transposed to index attention from/to appropriately. Surprisingly, KEY and QUERY appear to act somewhat differently, with QUERY being almost uniformly better than KEY with the best heads, whilst KEY is slightly better with averages, implying distinctions in how both store information. Furthermore, allowing both keys and queries seems to result in an interesting contradiction – the ultimate layer, which has reasonable maximums and averages for both KEY and QUERY, now seems to show a UUAS drop almost uniformly. This is also true for the completely unfrozen encoder.

Supervised Parsing In addition to decoding trees from attention matrices, we also measure supervised UAS/LAS on a held-out test set.[5] Based on Figure VII.4, it is apparent that all settings result in generally the same UAS. This is somewhat expected; Lauscher et al. (2020) see better results on parsing with the entire encoder frozen, implying that the task is easy enough for a biaffine parser to learn, given frozen mBERT representations.[6] The LAS distinction is, however, rather interesting: there is a marked difference between how important the dense layers are, as opposed to the attentive components. This is likely not reflected in our UUAS probe as, strictly speaking, labelling arcs is not equivalent to searching for structure in sentences, but more akin to classifying pre-identified structures. We also note that DENSE appears to be better than NONE on average, implying that non-dense components might actually be hurting labelling capacity.

[4] The inner average is over all heads; the outer is over all languages.

[5] Note that the test set in our scenario is from the actual, non-parallel language treebank; as such, we left Korean out of this comparison due to annotation differences.

[6] Due to training on concatenated PUD sets, however, our results are not directly comparable/

In brief, consolidating the two sets of results above, we can draw three interesting conclusions about the components:

1. **Value** vectors are best aligned with syntactic dependencies; this is reflected both in the best head at the upper layers, and the average score across all heads.
2. **Dense** layers appear to have moderate informative capacity, but appear to have the best learning capacity for the task of arc labelling.
3. Perhaps most surprisingly, **Key** and **Query** vectors do not appear to make any outstanding contributions, save for sharpening a smaller subset of heads.

Our last result is especially surprising for UUAS decoding. Keys and queries, fundamentally, combine to form the attention weight matrix, which is precisely what we use to decode trees. One would expect that allowing these components to learn from labelled syntax would result in the best improvements to decoding, but all three have surprisingly negligible mean improvements. This indicates that we need to further improve our understanding of how attentive structure and weighting really works.

Cross-linguistic observations We notice no clear cross-linguistic trends here across different component sets; however, certain languages do stand out as being particularly hard to decode from the fine-tuned parser. These include Japanese, Korean, Chinese, French and Turkish. For the first three, we hypothesise that tokenization clashes with mBERT's internal representations may play a role. Indeed, as we hypothesized in Section VII.3.2, it could be the case that the composition of CJK characters into gold tokens for Chinese and Japanese may degrade the representations (and their corresponding attention) therein. Furthermore, for Japanese and Korean specifically, it has been observed that tokenization strategies employed by different treebanks could drastically influence the conclusions one may draw about their inherent hierarchical structure (Kulmizev et al., 2020). Turkish and French are admittedly more difficult to diagnose. Note, however, that we fine-tuned our model on a concatenation of all PUD treebanks. As such, any deviation from PUD's annotation norms is therefore likely to be heavily penalised, by virtue of signal from other languages drowning out these differences.

VII.6 Conclusion

In this study, we revisited the prospect of decoding dependency trees from the self-attention patterns of Transformer-based language models. We elected to extend our experiments to 18 languages in order to gain better insight about how tree decoding accuracy might be affected in the face of (modest) typological diversity. Surprisingly, across all languages, we were able to decode dependency trees from attention patterns more accurately than an adjacent-linking baseline, implying that some structure was indeed being tracked by the mechanism. In looking at specific relation types, we corroborated previous studies in showing that particular layer-head combinations tracked the same relation with regularity across languages, despite typological differences concerning word order, etc.

In investigating the extent to which attention can be guided to properly capture structural relations between input words, we fine-tuned mBERT as input to a dependency parser. This, we found, yielded large improvements over the pretrained attention patterns in terms of decoding accuracy, demonstrating that the attention mechanism was learning to represent the structural objective of the parser. In addition to fine-tuning the entire mBERT network, we conducted a series of experiments, wherein we updated only select components of model and left the remainder frozen. Most surprisingly, we observed that the Transformer parameters designed for composing the attention matrix, K and Q, were only modestly capable of guiding the attention towards resembling the dependency structure. In contrast, it was the Value (V) parameters, which are used for computing a weighted sum over the KQ-produced attention, that yielded the most faithful representations of the linguistic structure via attention.

Though prior work (Kovaleva et al., 2019; Zhao and Bethard, 2020) seems to indicate that there is a lack of a substantial change in attention patterns after fine-tuning on syntax- and semantics-oriented classification tasks, the opposite effect has been observed with fine-tuning on negation scope resolution, where a more explanatory attention mechanism can be induced (Htut et al., 2019). Our results are similar to the latter, and we demonstrate that given explicit syntactic annotation, attention weights do end up storing more transparently decodable structure. It is, however, still unclear which sets of transformer parameters are best suited for learning this information and storing it in the form of attention.

Acknowledgements

Our experiments were run on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway, under the NeIC-NLPL umbrella. Mostafa and Anders were funded by a Google Focused Research Award. We would like to thank Daniel Dakota and Ali Basirat for some fruitful discussions and the anonymous reviewers for their excellent feedback.

References

Bahdanau, D. et al. (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: ed. by Bengio, Y. and LeCun, Y.

Belinkov, Y. and Glass, J. (2019). “Analysis Methods in Neural Language Processing: A Survey”. In: *Transactions of the Association for Computational Linguistics* vol. 7.

Chi, E. A. et al. (2020). "Finding Universal Grammatical Relations in Multilingual BERT". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Chu, Y.-J. (1965). "On the shortest arborescence of a directed graph". In: *Scientia Sinica* vol. 14.

Clark, K. et al. (2019). "What Does BERT Look at? An Analysis of BERT's Attention". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics.

Devlin, J. et al. (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)*.

Dozat, T. and Manning, C. D. (2016). "Deep biaffine attention for neural dependency parsing". In: *arXiv preprint arXiv:1611.01734*.

Hewitt, J. and Manning, C. D. (2019). "A structural probe for finding syntax in word representations". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory". In: *Neural computation* vol. 9, no. 8.

Htut, P. M. et al. (2019). "Do Attention Heads in BERT Track Syntactic Dependencies?" In: *arXiv preprint arXiv:1911.12246*.

Hu, J. et al. (2020). "XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization". In: *arXiv:2003.11080 [cs]*. arXiv: 2003.11080.

Jain, S. and Wallace, B. C. (2019). "Attention is not Explanation". 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)*.

Klein, D. and Manning, C. D. (2004). "Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kondratyuk, D. and Straka, M. (2019a). "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)*.

— (2019b). "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.

Kovaleva, O. et al. (2019). "Revealing the Dark Secrets of BERT". en. 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.

Kulmizev, A. et al. (2020). "Do Neural Language Models Show Preferences for Syntactic Formalisms?" In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Lauscher, A. et al. (2020). "From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Limisiewicz, T. et al. (2020). "Universal Dependencies according to BERT: both more specific and more general". In: *arXiv preprint arXiv:2004.14620*.

McDonald, R. et al. (2005a). "Non-Projective Dependency Parsing using Spanning Tree Algorithms". In: *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

McDonald, R. et al. (2005b). "Online Large-Margin Training of Dependency Parsers". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nivre, J. et al. (2016). "Universal Dependencies v1: A Multilingual Treebank Collection". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*.

Nivre, J. et al. (2019). *Universal Dependencies 2.4*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Nivre, J. et al. (2020). "Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection". In: *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*.

Pires, T. et al. (2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Raganato, A., Tiedemann, J., et al. (2018). "An analysis of encoder representations in transformer-based machine translation". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. The Association for Computational Linguistics.

Rocktäschel, T. et al. (2015). "Reasoning about entailment with neural attention". In: *arXiv preprint arXiv:1509.06664*.

Rosa, R. and Mareček, D. (2019). "Inducing syntactic trees from bert representations". In: *arXiv preprint arXiv:1906.11511*.

Serrano, S. and Smith, N. A. (2019). "Is Attention Interpretable?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Strubell, E. et al. (2018). "Linguistically-Informed Self-Attention for Semantic Role Labeling". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Swayamdipta, S. et al. (2018). “Syntactic Scaffolds for Semantic Structures”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Tenney, I. et al. (2019). “BERT rediscovers the classical NLP pipeline”. In: *arXiv preprint arXiv:1905.05950*.

Vaswani, A. et al. (2017a). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al. Vol. 30. Curran Associates, Inc.

— (2017b). “Attention is all you need”. In: *Advances in neural information processing systems*.

Voita, E. et al. (2019). “Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Wiegreffe, S. and Pinter, Y. (2019). “Attention is not not Explanation”. 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)*.

Wu, Y. et al. (2016). “Google’s neural machine translation system: Bridging the gap between human and machine translation”. In: *arXiv preprint arXiv:1609.08144*.

Xu, K. et al. (2015). “Show, attend and tell: Neural image caption generation with visual attention”. In: *International conference on machine learning*.

Zeman, D. et al. (2018). “CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies”. In: *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*.

Zhao, Y. and Bethard, S. (2020). “How does BERT’s attention change when you fine-tune? An analysis methodology and a case study in negation scope”. en. In.

Appendix VII.A Positional Scores Per Offset

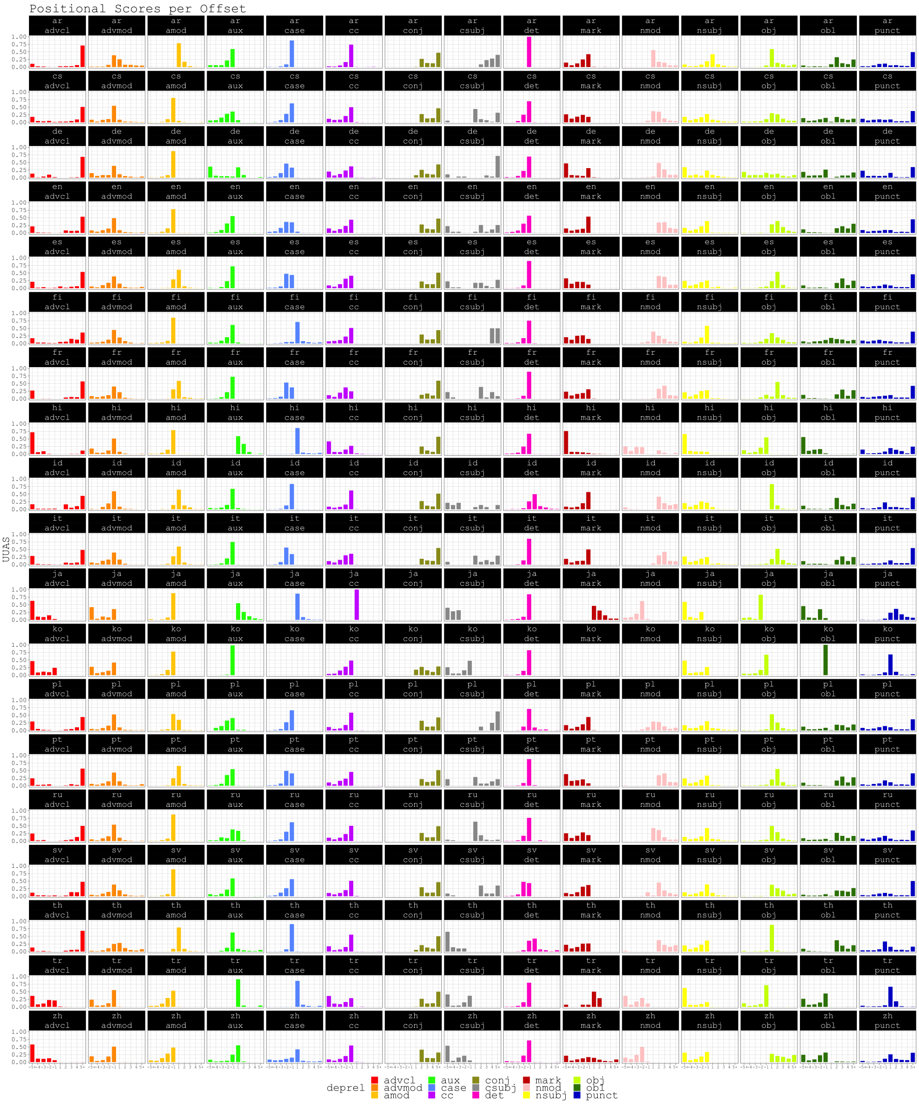


Figure VII.5: Positional scores across relations for all languages.

Appendix VII.B Decoding UUAS Across Relations

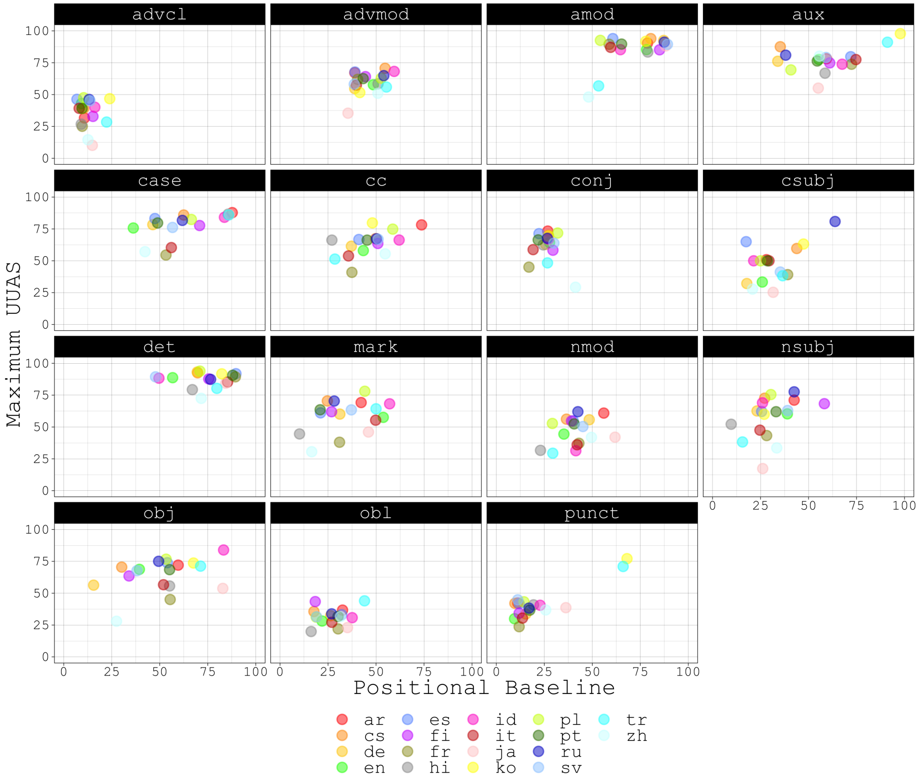


Figure VII.6: Decoding UUAS as a function of best positional baselines.

Appendix VII.C Full Parsing Scores

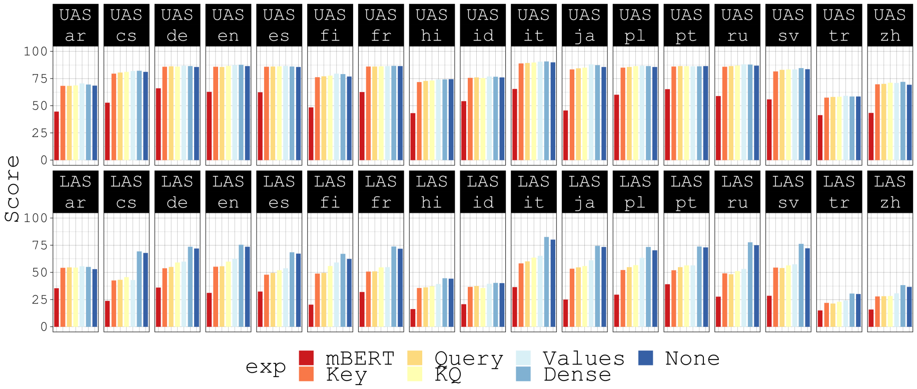


Figure VII.7: Parsing scores across components and languages.

Paper VIII

The Impact of Positional Encodings on Multilingual Compression

Vinit Ravishankar, Anders Søgaard

Appears in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, November 2021, pp. 763–777. DOI: 10.18653/v1/2021.emnlp-main.59.

Abstract

In order to preserve word-order information in a non-autoregressive setting, transformer architectures tend to include positional knowledge, by (for instance) adding positional encodings to token embeddings. Several modifications have been proposed over the sinusoidal positional encodings used in the original transformer architecture; these include, for instance, separating position encodings and token embeddings, or directly modifying attention weights based on the distance between word pairs. We first show that surprisingly, while these modifications tend to improve monolingual language models, none of them result in better multilingual language models. We then answer why that is: Sinusoidal encodings were explicitly designed to facilitate compositionality by allowing linear projections over arbitrary time steps. Higher variances in multilingual training distributions requires higher compression, in which case, compositionality becomes indispensable. Learned absolute positional encodings (e.g., in mBERT) tend to approximate sinusoidal embeddings in multilingual settings, but more complex positional encoding architectures lack the inductive bias to effectively learn compositionality *and* cross-lingual alignment. In other words, while sinusoidal positional encodings were originally designed for monolingual applications, they are particularly useful in multilingual language models.

Contents

VIII.1 Introduction

Multiple recent papers have attempted to pinpoint precisely what components of multilingual language models enable cross-lingual transfer. Pires et al. (2019) show that although wordpiece overlap tends to improve cross-lingual transfer performance, even languages with different scripts (and no shared subwords) may enable zero-shot transfer. Wu and Dredze (2019) report similar results on a wider range of tasks. Artetxe et al. (2020) show that neither a shared vocabulary nor joint multilingual pre-training are necessary to train successful multilingual models. K et al. (2020) find that model depth is a contributor to transfer performance, but that reducing the number of self-attention heads does not have much of an effect.

Our starting point is Dufter and Schütze (2020), who claim that a) multilingual compression is caused by forced parameter sharing across languages, and that b) positional encodings play a significant role in the creation of a multilingual space, even in the absence of shared subwords and shared special tokens, like delimiters.

Contributions We build on Dufter and Schütze (2020) and demonstrate, through a series of experiments on synthetic and real data, that the choice of positional encoding mechanism has a significant effect on cross-lingual model performance: While many positional encodings have been proposed in monolingual settings as improvements over sinusoidal or absolute positional encodings, originally proposed in Vaswani et al. (2017) and Devlin et al. (2019), including untied positional encodings (TUPE; Ke et al. (2020)) and relative positional encodings Huang et al., 2020; Shaw et al., 2018, none of these better facilitate cross-lingual compression or sharing. In fact, multilingual language models trained with untied or relative positional encodings exhibit *much worse* cross-lingual performance. We show that this is because sinusoidal embeddings

Sinusoidal	See §2	Vaswani et al. (2017)
Absolute	$((w_i + p_i)W^{Q,1})$ $((w_j + p_j)W^{K,1})^\top$	Devlin et al. (2019)
TUPE	$(x_i^l W^{Q,l})(x_j^l W^{K,l})^\top +$ $(p_i U^Q)(p_j U^K)^\top$	Ke et al. (2020)
TUPE(r)	$\ldots + b_{j-i}$	
Relative(k)	$(x_i W^Q)(x_j W^K + a_{ij})^\top$	Shaw et al. (2018)
Relative(k/q)	$(x_i W^Q + a_{ij})$ $(x_j W^K + a_{ij})^\top$	Huang et al. (2020)

Table VIII.1: We compare six positional encodings and their impact on cross-lingual generalization in multilingual language models

facilitate compositionality, which we argue is particularly important for cross-lingual compression. We present a method for quantifying the compositionality of positional encodings, and find additional evidence for this hypothesis in word-position correlations and ablation studies. We are, to the best of our knowledge, the first to show this asymmetry between monolingual and multilingual language model training. Our experiments rely on the protocols in Dufter and Schütze (2020), but in addition to simple experiments with their Bible data, we also replicate all our experiments on Wikipedia data. Rather than relying on deterministic perturbations of data, as in Dufter and Schütze (2020) and Sinha et al. (2021), we make novel use of Galactic Dependencies Wang and Eisner, 2016 in our experiments. Based on our experiments, we recommend caution when adopting methods developed for monolingual language models when training multilingual models, as well as that future work on positional encoding mechanisms also provides evaluations in multilingual settings.

VIII.2 Positional encodings

Positional encodings have been a mainstay of non-autoregressive transformer-based models right since Vaswani et al. (2017) first proposed the transformer architecture. The motivation being that given that transformers[1] are order-invariant (as opposed recurrent or convolutional networks), there must be some injection of word order into the encoder. Rather than using conventional "embeddings", Vaswani et al. (2017) use fixed **sinusoidal** position encodings, where each dimension characterises a sinusoidal waveform of a fixed frequency. Specifically, each encoding p is given as:

$$\begin{aligned} p_{(pos,2i)} &= sin(pos/10000^{2i/d_{\text{model}}}) \\ p_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{\text{model}}}) \end{aligned}$$

where pos is the position and i is the dimension. They add these encodings to token representations before passing the sum to the first layer of the self-attention mechanism.

Several alternatives to sinusoidal encodings have been proposed since Vaswani et al. (2017). Most multilingual models tend to use BERT-style (Devlin et al., 2019) learnt **absolute** positional encodings, where a unique vector is learned and assigned to each position; these vectors are then added to word representations before being passed to the self-attention mechanism.

As an alternative to such position representations, where every position is represented by a unique vector, **relative** positional encodings have been proposed (Huang et al., 2020; Shaw et al., 2018). Rather than assigning representations to tokens based on their position, relative positional encoding involves assigning representations to position-position pairs; typically, these encodings are calculated separately and added to the attention matrix. We

[1] Note that we use "transformers" as shorthand for transformer encoders used for masked language modelling.

evaluate both the encodings proposed in Shaw et al. (2018) and the encodings proposed in Huang et al., 2020 in our experiments below.

He et al. (2021) propose eliminating position-position correlations, and using separate parameters for word and position representations; Wang et al. (2019) propose using dependency trees instead of raw sequential positions. Ke et al. (2020) recommend eliminating the addition operation in BERT-style representations; they argue that word-position correlations are effectively nil, and that the addition introduces unnecessary noise. We evaluate two **untied** positional encodings proposed in Ke et al. (2020) (TUPE). TUPE modifies absolute representations by a) untying word-position correlations; b) using a separate set of parameters for positional attention and c) untying `[CLS]` tokens from positions.

We refer to recent surveys (Dufter et al., 2021; Wang et al., 2021) for a more detailed treatment of position encoding methods. We provide a summary of our methods in Table VIII.1. $W^{Q,l}$ and $W^{K,l}$ represent the query/key weights for the attention mechanism at some layer l, and a_{ij} or b_{j-i} are learnt vectors corresponding to the offset $j-i$. Note that the untied position-position term $(p_iU^Q)(p_jU^K)^\top$ is added at every layer.

The above positional encodings have been introduced in the context of monolingual pretrained language models, and there has been only a limited amount of work addressing the effect of positional encodings on multilingual models. Liu et al. (2020a) find that positional information tends to hurt machine translation, as the encoder learns a word-order bias towards the source languages.[2] Artetxe et al. (2020) find that language-specific positional representations help in an adapter-based training scenario. Ding et al. (2020) attempt to account for structural differences between languages by using bracketing transduction grammar trees to reorder position labels (and find that it helps). Liu et al. (2020b) find that models that are relatively agnostic to word-order tend to perform better in cross-lingual settings; they hypothesise that large multilingual encoders, being trained on languages with drastic differences in word orders, tend to have order-agnostic positional encodings, and thus discourage fine-tuning positional encodings downstream. Contemporaneous with this work, Sinha et al. (2021) show that positional information is important for monolingual models even given unnatural, randomly shuffled word ordering.

Dufter and Schütze (2020) present a set of experiments training smaller language models on bilingual corpora, consisting of the same corpus in English and "fake-English", which is English with a shifted BPE vocabulary. They evaluate retrieval and translation scores at different layers; gold alignments are easy to derive given that the corpora are effectively parallel corpora, and that the vocabularies for both halves are effectively the same. As we build on these

[2] The results in Liu et al. (2020a) apply to zero-shot generalization of fine-tuned, task-specific models and not to how multilingual language models are pretrained. In their experiments, they rely on a pretrained language model with absolute positional encodings. In fact, what they show is that freezing these during fine-tuning helps cross-lingual zero-shot generalization.

experiments, we adopt slightly simplified notation, and denote vocabulary-shifted corpora with square brackets, eg. [EN].

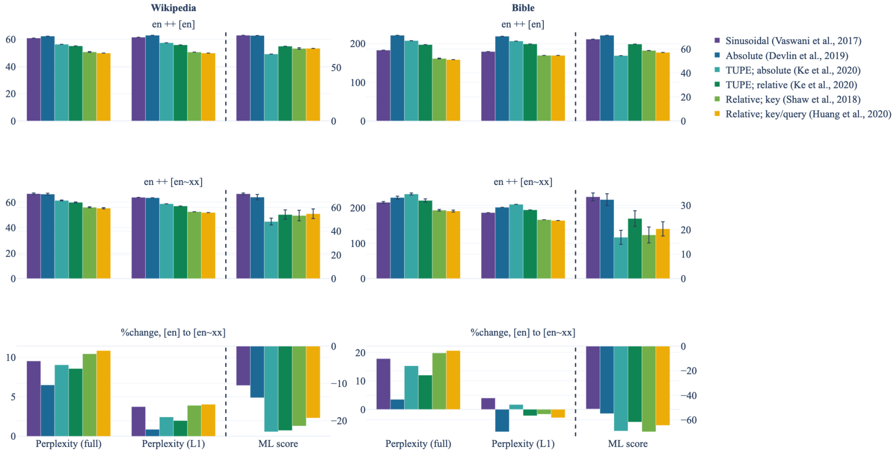


Figure VIII.1: **Main results:** While untied and relative positional encodings are superior to sinusoidal and absolute positional encodings in the monolingual setting, they are clearly worse in the multilingual setting, especially for structurally different languages. The multilingual (ML) scores are computed as in Dufter and Schütze (2020). Note also that results are averages across seven different word orders (see §3).

VIII.3 Experiments

Galactic Dependencies A drawback of the multilingual experiments presented in Dufter and Schütze (2020) is that EN and [EN] effectively have the same structure. While the authors attempt to control for this in additional experiment where word order in [EN] is completely reversed, this does not resemble realistic differences across languages. Using true multilingual corpora is, however, difficult: our retrieval and translation tasks are easy to bootstrap precisely because we have faux-parallel corpora, with effectively pre-aligned vocabulary.

To induce structural diversity in our corpora, therefore, we reorder our corpora using Galactic Dependencies (GD) models (Wang and Eisner, 2016). Briefly, GD models sample ordering statistics based on dependency relations for the dependants of verbs and/or nouns from some superstrate language XX; when applied to sentences in some substrate language (in the context of our experiments, EN), the models reorder dependants of VERB and/or NOUN nodes to match the ordering statistics of the substrate language they were trained on. We opt to reorder both nominal and verbal arguments, and follow the authors in denoting the sampling operation with a $\sim$, giving us for eg. EN$\sim$XX for an English language corpus, with dependent order statistics adapted from some language XX. Table VIII.2 contains an example sentence and some of its reorderings.

Note that GD reordering only works for projective sentences, and rather than retain un-reordered non-projective sentences, we exclude them from all our corpora.

EN	So there were fourteen generations from Abraham to David .
EN∼AR	. there were So generations fourteen from Abraham to David
EN∼DE	there were So from Abraham to David fourteen generations .
EN∼EU	there were So David to Abraham from generations fourteen .
EN∼FI	Abraham from David to fourteen generations there were So .
EN∼FR	fourteen generations from Abraham to David were there So .
EN∼HI	there So David to Abraham from fourteen generations were .
EN∼SV	there were So generations from Abraham to David fourteen .

Table VIII.2: An example sentence from the easy-to-read Bible with its GD reorderings.

This approach, while simple and useful, does have several limitations. Predominantly, because our reordering is fundamentally syntactic/structural, our fake languages still maintain both the morphology of the source language (English in our case), and the same vocabulary distribution. Thus, although scrambling ought to affect context and neighbourhoods, an English token and its corresponding fake token have exactly the same unigram distribution.

Training Our model of choice is an underparameterised BERT, as in Dufter and Schütze (2020). We train multiple such underparameterised BERT models, each with a different encoding mechanism from Section VIII.2, on two bilingual corpora:

en + [en] - a bilingual corpus comprised of English, and a fake vocab-shifted English.

en + [en∼xx] - a bilingual corpus comprised of English, and a fake English that has had its constituents reordered to match the distribution of some language XX.

We reorder our English starting point according to seven different faux-languages (just "languages" for brevity): Arabic, German, Basque, Finnish, French, Hindi and Swedish. Note that given that our starting point was English, there was no way for us to control for morphological differences; as such, languages with freer word order (like Basque) are likelier to make our English corpora ambiguous.

We use two corpora in this work: the first is the Bible splits from Dufter and Schütze (2020), with the English easy-to-read Bible as the training split, and the KJV Bible as validation. The second corpus uses the English Wikipedia as the training split, and Common Crawl as validation. We present corpus statistics in Table VIII.3. For each corpus, we learn and apply a BPE vocabulary of size 2048.

Following Dufter and Schütze (2020), our BERT models all have a single head and 12 layers. We reduce the dimensionality of the encoder layers to 64,

	Train	Validation
Bible	30602	9080
Wikipedia	50000	20000

Table VIII.3: Corpus sizes in sentences (two languages per corpus)

and the feed-forward layers to 256. Each model is trained for 100 epochs with three different random seeds (0, 42 and 100), giving us a total of 7 languages x 6 encoding methods x 3 seeds x 2 corpora = 252 models. We implement our code[3] in the transformers library (Wolf et al., 2020). For learned absolute and the two relative encoding models, we use the default implementations, that scale attention operations by a scaling factor of $\frac{1}{\sqrt{d}}$. For our untied models, we adjust our scaling factor to $\frac{1}{\sqrt{2d}}$ as in the original paper (Ke et al., 2020). For sinusoidal representations, while Vaswani et al. (2017) multiply token embeddings by $\sqrt{d}$ to avoid drowning them out with the $[-1, 1]$ sinusoidal encoding range, we find that our default embedding size is too small for this to have an effect, and instead scale up token embeddings by $2\sqrt{d}$ before adding positional encodings.

For all parameterised encoding models except TUPE (relative), we use a maximum of $k = 512$ positions; the concrete transformers implementation of the relative methods means that this gives us 1023 total offsets. [4] For TUPE (relative), we use a maximum of $k = 128$ positions, divided into 32 bins with logarithmically increasing bin sizes; this is taken from the original implementation in Ke et al. (2020).

VIII.4 Evaluation

We adopt Dufter and Schütze's e (2020)valuation pipeline, evaluating each of our models at layers 0 and 8; we also describe a multilingual score, which is defined as the average accuracy for the retrieval and translation tasks, at layers 0 and 8. We also measure perplexity, both on the monolingual first half of the corpus, and on both halves combined. Note that true perplexities for masked language models are intractable (Salazar et al., 2020; Wang and Cho, 2019). We use a trivial approximation and calculate perplexity based on the prediction loss for each masked token; note that while these suffice for comparison purposes, they are not true perplexities and should not be taken as such outside the context of these experiments.

We present our results (averaged out over faux-languages) in Figure VIII.1, with full results in Appendix VIII.C. As expected, the more recent positional encodings are superior to sinusoidal or absolute positional encodings in the monolingual setting; but somewhat surprisingly, sinusoidal and absolute positional encodings are clearly outperforming the more recent approaches in

[3] github.uio.no/vinitr/multilingual-position

[4] In line with Shaw et al. (2018), we also attempted to use $k = 16$ for the relative key model, but saw no difference in results.

	Wiki/CC							Bible						
Embedding	Perplexity		Retrieval		Translation		ML score	Perplexity		Retrieval		Translation		ML score
	Full	L1	0	8	0	8		Full	L1	0	8	0	8	
Sinusoidal	66.73	63.96	37.43	**97.29**	**77.03**	**64.07**	**68.95**	215.77	186.53	4.82	**54.09**	47.09	**27.62**	**33.4**
Absolute	66.35	63.44	**52.35**	96.53	76.05	53.62	68.59	229.28	201.88	**9.62**	52.51	**47.6**	19.36	32.27
TUPE (absolute)	61.4	58.77	9.61	84.72	65.89	36.57	48.07	239.48	210.16	1.65	28.86	28.85	8.12	16.87
TUPE (relative)	59.81	56.96	16.25	88.5	71.7	40.54	53.89	221.04	194.58	2.54	41.34	40.39	13.92	24.55
Relative (key)	55.98	52.5	20.2	87.36	73.09	31.38	53.09	193.49	166.75	2.18	28.46	30.43	10.25	17.83
Relative (key/query)	**55.28**	**51.83**	21.24	88.04	73.58	34.42	54.64	**191.23**	**164.41**	2.4	31.6	34.26	13.03	20.32

Table VIII.4: Detailed results, averaged across our faux-languages. Best results per metric in bold.

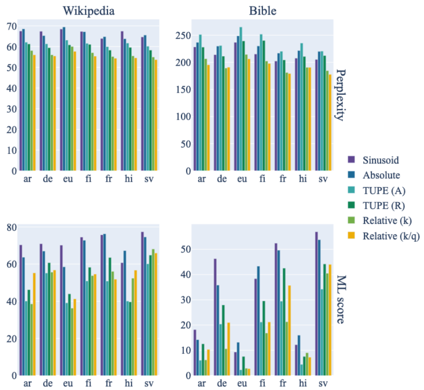


Figure VIII.2: (Full) perplexity and ML score across languages.

the multilingual setting. We also note that the gap in multilingual performance only grows larger when a different word order is imposed on the target language; see the bottom row of Figure VIII.1. Interestingly, switching to structurally different L2s can sometimes reduce the language modelling perplexity of the L1: this could be due to regularisation induced by structural differences.

Typological differences We discuss "typology" with a caveat: our experiments with GD only alter word order, which means that all our altered-structure experiments still have English morphology. As such, it is impossible to talk about non-English languages; only about non-English word-order tendencies, when induced in English. Having said that, when we measure performance variation across languages (Figure VIII.2), our results are more or less what one would expect: performance is decent for relatively rigid word-order languages, and poorer for languages that have complex morphology.

Interestingly, SVO languages consistently tend to perform better than our three non-SVO languages (Basque, Hindi and Arabic); this could be due to VSO/SOV languages requiring morphology to disambiguate between adjacent nominals (Levshina, 2019). Another justification could also be that these are languages with a very different "default" word order to English; this would

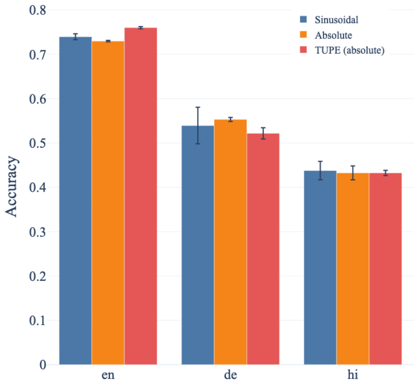


Figure VIII.3: Real-world results on XNLI. Models were pretrained on a large text corpus and finetuned on English MultiNLI.

further motivate Ding et al.'s u (2020)se of cross-lingually reordered position markers.

Real-world results While we conduct most of our analyses on our toy models, we also ran a series of experiments to verify that our results would hold with larger models. As such, we pre-trained full size BERT models (base, not large) for two epochs, on a corpus consisting of 8.5M, 9.3M and 800k sentences in English, German and Hindi respectively. We then fine-tuned these models for three epochs on (English) MultiNLI (Williams et al., 2018), and evaluated on held-out XNLI test sets for our three languages (Conneau et al., 2018); the process took approximately 4 days per model, on a single V100 GPU. We trained two models (seeds 0 and 42) per method, for three different positional encoding methods: a) absolute positional encodings, as these are used in the original BERT, b) sinusoidal encodings, as these were the original transformer encodings, and c) TUPE (absolute), as the most recent innovation. Our real-world results appear to validate our toy experiments: performance on English, the language the model was fine-tuned on, is highest with TUPE, while cross-lingual transfer suffers, both on German and to a lesser extent on Hindi.

VIII.5 Analyses

In an attempt to explain the significantly improved cross-lingual performance of absolute positional encodings, we tried to examine precisely what sort of encoding was being learnt. Part of the original motivation behind sinusoidal encodings was that they would allow for **compositionality**; for any fixed offset

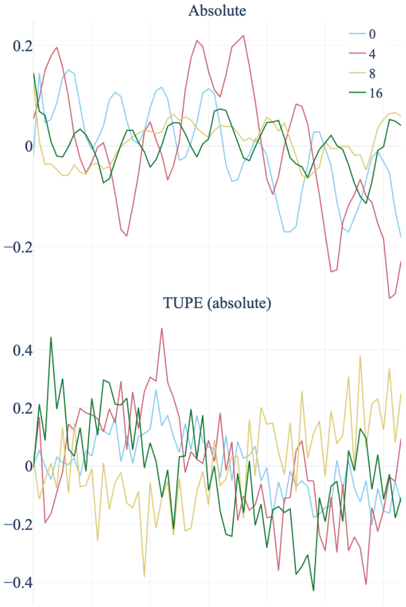


Figure VIII.4: Dimensions 0, 4, 8 and 16 of learnt absolute and TUPE positional encodings over 32 positions for EN∼FI, seed 0.

k, there exists a linear transformation from p_{pos} to p_{pos+k}, making it easier to learn to attend to relative offsets; the proof of this is in Appendix VIII.A.[5]

We examined our absolute positional encodings to see whether or not they were being induced to learn some specific function. Figure VIII.4 plots 4 dimensions of absolute and TUPE(a) positional encoding, for the EN + [EN∼FI] model; each line represents a specific dimension of the encoding vectors generated for positions 0 to 31. Interestingly, it appears that absolute representations converge to waveforms that represent sinusoids somewhat, while neither of the untied experiments do so (cf. Appendix VIII.B).

We hypothesize that absolute representations converge to waveforms because of increased pressure for compositionality, being trained on structurally different languages. To test this, we quantify the extent to which the absolute, relative and untied encodings are compositional in the sense that there is a linear transformation from p_{pos} to p_{pos+k} for different k.

To this end, we use Procrustes analysis Stegmann and Gomez, 2002 to learn a linear transformation for each k, based on the representations of p_{pos} and p_{pos+k}.

[5] Vaswani et al. (2017) do not explicitly mention compositionality, but only generalization across positions for fixed offsets. Positional disentanglement is the flipside of compositionality, however Chaabouni et al., 2020.

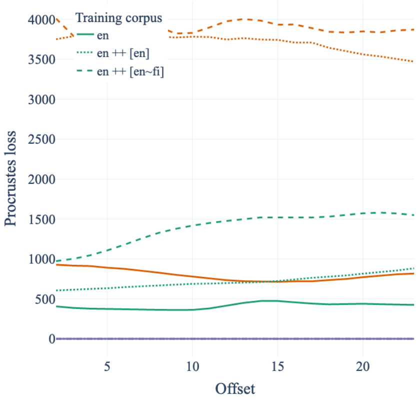


Figure VIII.5: Procrustes loss for absolute encodings and TUPE (seed 0); differences are statistically significant with $p < 0.001$ (Wilcoxon). The sinusoidal loss is ≈ 0.

Specifically, we apply *orthogonal* Procrustes analyses (Schönemann, 1966), which avoid scaling and translation.

First, we minimise $\arg\min_{\mathrm{T}} ||p_{pos} - \mathrm{T}p_{pos+k}||^2$. Next, we apply T to a different randomly selected pos', i.e. we calculate $\mathcal{L} = ||p_{pos'} - \mathrm{T}p_{pos'+k}||^2$. The higher the final loss $\mathcal{L}$, the less our encodings facilitate compositionality. In order to make learning T simpler, rather than selecting representations for single positions pos and pos', we select chunks of arbitrary size C, and stack their positions into a matrix. Note that for sinusoidal representations, the loss is close to zero regardless of span.

The losses are plotted over a range of offsets for both absolute representations and for TUPE(a), in Figure VIII.5; we include a control model trained on a monolingual corpus. Losses are averaged over 125 runs per offset, with random values of pos, pos' and C. While both forms of representation appear to be similar (and relatively non-sinusoidal) when trained on the monolingual corpus, introducing bilingualism leads to a clear difference between the two: absolute positional representations tend to be a lot closer to sinusoidal representations than untied ones do. Note, also, that this gap is clearest for the (simpler) EN + [EN] experiment – this is unsurprising, as EN + [EN] is still *perceived* as bilingual due to the shifted vocabulary. The structural similarity between the two, however, makes it easier to build compositional representations by relying on offsets, as the model only needs to learn to represent one language, structurally speaking. We observe a similar gap when comparing pretrained BERT models: bert-base-multilingual-cased exhibits more sinusoidal representations over a range of offsets, when compared to bert-base-cased, although the gap is narrower than with our toy models.

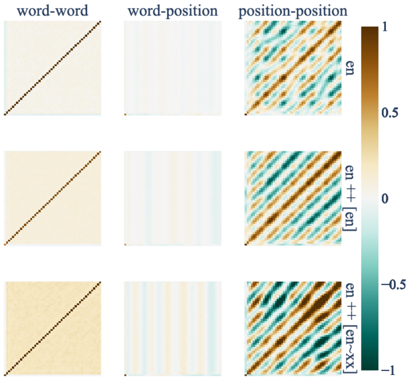


Figure VIII.6: Word-position correlations for our Finnish-reordered model with random seed 0; words on the y- and positions on the x-axis.

Correlations in multilingual settings A key motivation for eliminating word-position correlations, presented in (Ke et al., 2020), is the fact that these correlations are effectively zero, leading to no additional information for the model. Figure VIII.6 captures word-position correlations from three of our trained models (with an additional model trained on a purely monolingual corpus); note that while these correlations are very close to zero for monolingual corpora, there is a visible "banding" phenomenon in the multilingual corpora, that only grows stronger when a different grammar is sampled. A similar banding phenomenon is visible when we compare multilingual and monolingual pre-trained BERT models (Appendix VIII.B), albeit with reduced magnitude. We hypothesize that the pressure for compositionality induces these correlations.

Ablation studies Finally, we ran a series of ablation experiments on absolute positional encodings to support the above analysis. Three of the experiments involved removing position-position correlations, position-word correlations, word-position correlations, and a fourth involved using separate parameters for word and position attention. Results are presented in Figure VIII.7; we also include the median Procrustes loss. We note that the removal of both position-word correlations and word-position correlations has an effect on both perplexity and ML score. Interestingly, removing word-position correlations ($(p_iW^Q)(w_jW^K)^\top$) does not have the same effect as the inverse does: perplexity is lower than with position-word correlations removed, but so is the ML score, indicating a difference between the role played by position as a key, and as a query.

On relative representations Given our previous assumptions about offsets aiding compositionality, why, then, do our relative representations - that explic-

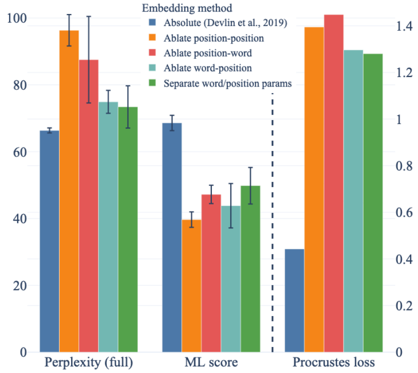


Figure VIII.7: Ablation experiments, averaged over languages (for perplexity and ML score). Procrustes losses calculated as in §5, for the EN∼FI model (seed 0).

itly calculate offsets - perform poorly in multilingual settings? We speculate that the reason relative encodings appear to hurt multilingual compression is that offset-specific bias terms sparsify the learning signal for (and thereby hinder the alignment of) disjoint vocabularies. In compensating for this, relative positional encodings sacrifice their compositionality. Relative representations aid compositionality by directly providing a bias term derived from the distance between a word pair. As shown above, absolute representations learn similar biases; however, being actively forced to learn such biases could encourage models to jointly learn alignment and compositionality.

Further, offset representations are also effectively "hard", i.e. derived from the hard distance between the two tokens. The interaction between w_i and w_j is not wholly mediated by the distance $i - j$, however, this correlation is forced by the product term $(x_i W^Q)(a_{ij})^\top$. The term $(x_i W^Q)(p_j W^K)^\top$, on the other hand, could effectively attend to multiple offsets. $p_j W^K$ is fixed for position j; given the sinusoidal nature of p, the product term could induce a "soft" positional representation with subspaces attending to different offsets[6]; the relevant offset mix could then be indexed into by $x_i W^Q$.

VIII.6 Discussion

The main contribution of our work is practical, namely showing that findings about positional encodings in the context of monolingual language models do not apply straightforwardly to multilingual language models. In answering why sinusoidal embeddings are superior to more recent alternatives in the

[6] Indeed, we find that $p_j W^k$ is less invariant to Procrustes transformation than p_j is.

multilingual setting, we also found the compositionality of positional encodings to be predictive of multilingual compression in such models. While relative positional encodings seem designed for compositionality, they prevent efficient alignment of multilingual vocabularies.

Sinha et al. (2021) show that word order matters little for monolingual language model pretraining, and that pretrained language models seem to rely mostly on higher-order word co-occurrence statistics. Our work shows that this finding does not generalize to pretraining multilingual language models. In the multilingual setting, word order clearly matters, as also shown in previous work (Dufter and Schütze, 2020; Ke et al., 2020), and compositional positional encodings seem to facilitate effective multilingual compression. This aligns with the observation that syntactic reordering à la Ding et al. (2020) is in some cases an effective way to encourage compositional cross-lingual representations.

In general, our results illustrate how methods developed for monolingual language models should not be blindly adopted when training multilingual models, which potentially require different architectures. Conversely, we would encourage future work on new positional encoding mechanisms for non-autoregressive models to also evaluate these mechanisms in multilingual settings.

VIII.7 Conclusion

Through a series of synthetic and real experiments with training multilingual language models, we showed that a) sinusoidal positional encodings perform better in multilingual settings than more recent alternatives (that have been shown to perform better in monolingual settings); b) this is likely because of an increased pressure for compositionality. We devised a method for quantifying the compositionality of positional encodings, and strengthened our results by also considering word-position correlations and ablation studies.

Acknowledgements

We would like to thank Lilja Øvrelid, Erik Velldal, Mostafa Abdou and Artur Kulmizev for their feedback and comments. We would also like to acknowledge the supercomputing services provided by the CSC IT Center for Science in Finland.

References

Artetxe, M. et al. (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.

Chaabouni, R. et al. (2020). “Compositionality and Generalization In Emergent Languages”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Conneau, A. et al. (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.

Devlin, J. et al. (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.

Ding, L. et al. (2020). “Self-Attention with Cross-Lingual Position Representation”. In: *arXiv:2004.13310 [cs]*.

Dufter, P. and Schütze, H. (2020). “Identifying Elements Essential for BERT’s Multilinguality”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Dufter, P. et al. (2021). “Position Information in Transformers: An Overview”. In: *arXiv:2102.11090 [cs]*.

He, P. et al. (2021). “DeBERTa: Decoding-Enhanced BERT with Disentangled Attention”. In: *International Conference on Learning Representations*.

Huang, Z. et al. (2020). “Improve Transformer Models with Better Relative Position Embeddings”. In.

K, K. et al. (2020). “Cross-Lingual Ability of Multilingual BERT: An Empirical Study”. In: *International Conference on Learning Representations*.

Ke, G. et al. (2020). “Rethinking Positional Encoding in Language Pre-Training”. In: *International Conference on Learning Representations*.

Levshina, N. (2019). “Token-Based Typology and Word Order Entropy: A Study Based on Universal Dependencies”. In: *Linguistic Typology* vol. 23, no. 3.

Liu, D. et al. (2020a). “Improving Zero-Shot Translation by Disentangling Positional Information”. In: *arXiv:2012.15127 [cs]*.

Liu, Z. et al. (2020b). “On the Importance of Word Order Information in Cross-Lingual Sequence Labeling”. In: *arXiv:2001.11164 [cs]*.

Pires, T. et al. (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.

Salazar, J. et al. (2020). “Masked Language Model Scoring”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Schönemann, P. H. (1966). “A generalized solution of the orthogonal procrustes problem”. In: *Psychometrika* vol. 31, no. 1.

Shaw, P. et al. (2018). “Self-Attention with Relative Position Representations”. In.

Sinha, K. et al. (2021). “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-Training for Little”. In: *arXiv:2104.06644 [cs]*.

Stegmann, M. B. and Gomez, D. D. (2002). *A Brief Introduction to Statistical Shape Analysis*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Vaswani, A. et al. (2017). "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al. Vol. 30. Curran Associates, Inc.

Wang, A. and Cho, K. (2019). *BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model*. arXiv: 1902.04094 [cs.CL].

Wang, B. et al. (2021). "ON POSITION EMBEDDINGS IN BERT". In.

Wang, D. and Eisner, J. (2016). "The Galactic Dependencies Treebanks: Getting More Data by Synthesizing New Languages". In: *Transactions of the Association for Computational Linguistics* vol. 4.

Wang, X. et al. (2019). "Self-Attention with Structural Position Representations". 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.

Williams, A. et al. (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics.

Wolf, T. et al. (2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics.

Wu, S. and Dredze, M. (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.

Appendix VIII.A Proof of linear transformability

Let

$$\begin{bmatrix}\sin(\omega t)\\ \cos(\omega t)\end{bmatrix}$$

represent a sine/cosine pair, characterised by position t. Let

$$\mathrm{R}_k = \begin{bmatrix}\cos(\omega k) & \sin(\omega k)\\ -\sin(\omega k) & \cos(\omega k)\end{bmatrix}$$

be a rotation matrix for angle ωk. We then have:

$$R\begin{bmatrix}\sin(\omega t)\\ \cos(\omega t)\end{bmatrix} = \begin{bmatrix}\cos(\omega k) & \sin(\omega k)\\ -\sin(\omega k) & \cos(\omega k)\end{bmatrix}\cdot\begin{bmatrix}\sin(\omega t)\\ \cos(\omega t)\end{bmatrix}$$

$$= \begin{bmatrix} \sin(\omega k)\cos(\omega t) + \cos(\omega k)\sin(\omega t) \\ \cos(\omega k)\cos(\omega t) - \sin(\omega k)\sin \omega t) \end{bmatrix}$$

$$= \begin{bmatrix} \sin(\omega(t+k)) \\ \cos(\omega(t+k)) \end{bmatrix}$$

implying that for a fixed frequency ω, there exists a rotation matrix R_k that can induce a rotational offset of k.

Appendix VIII.B Additional plots

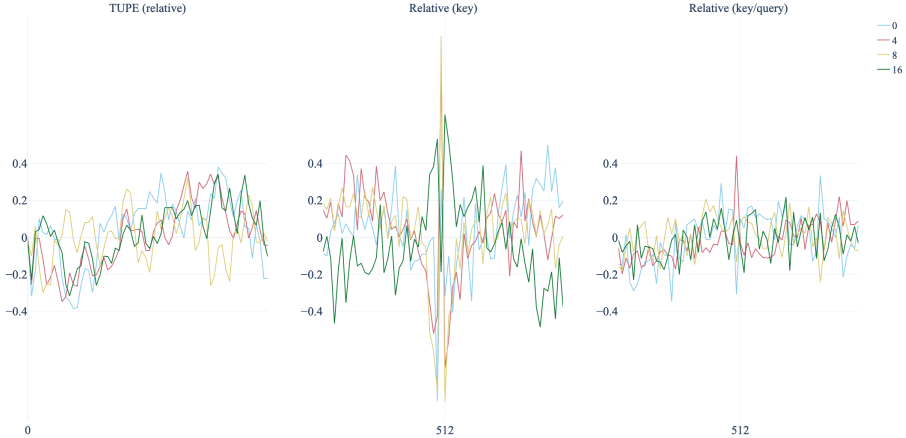


Figure VIII.8: Four neurons over 32 for TUPE (relative); the same neurons for 32 offsets centred on 512 for the other relative models.

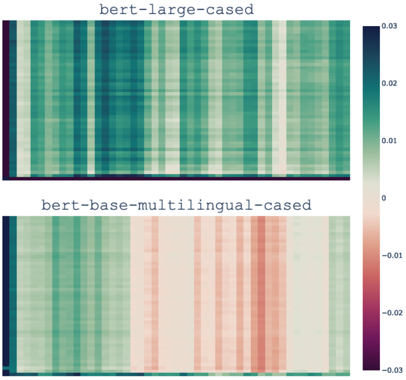


Figure VIII.9: Word-position correlations for pretrained BERT models

Appendix VIII.C Full results

Wiki/CC	Embedding	Perplexity		Retrieval		Translation		ML score
		Full	L1	0	8	0	8	
Arabic	Sinusoidal	67.52 (1.72)	64.16 (2.14)	47.90 (25.39)	98.09 (0.90)	71.75 (7.39)	62.52 (0.09)	70.36 (5.73)
	Absolute	68.54 (2.01)	65.02 (2.83)	44.84 (8.00)	95.14 (0.98)	70.79 (3.62)	46.74 (10.01)	63.67 (3.37)
	TUPE (a)	62.09 (0.57)	59.01 (0.45)	7.88 (1.63)	75.17 (7.47)	58.06 (4.65)	32.13 (3.28)	40.15 (4.45)
	TUPE (r)	61.22 (0.50)	57.81 (1.05)	15.12 (4.39)	83.44 (13.58)	64.94 (9.55)	29.03 (10.33)	46.22 (10.09)
	Relative (k)	58.08 (0.98)	53.92 (0.89)	12.61 (5.00)	74.14 (15.52)	63.27 (13.04)	9.37 (3.85)	38.56 (9.24)
	Relative (k/q)	55.96 (1.46)	51.98 (1.02)	20.95 (8.34)	89.53 (8.39)	73.01 (5.47)	35.83 (20.20)	55.23 (12.46)
German	Sinusoidal	67.34 (1.52)	64.74 (1.21)	47.12 (25.19)	98.94 (0.22)	71.53 (14.37)	61.06 (10.25)	70.95 (3.74)
	Absolute	65.28 (1.69)	62.67 (2.08)	63.68 (2.44)	98.87 (0.06)	80.21 (0.60)	36.34 (24.86)	67.01 (8.44)
	TUPE (a)	61.29 (0.70)	58.70 (0.60)	12.11 (1.29)	92.87 (2.50)	71.39 (3.10)	48.28 (3.36)	55.15 (2.09)
	TUPE (r)	59.42 (0.77)	56.90 (1.29)	18.27 (3.33)	97.40 (1.13)	77.86 (2.10)	43.72 (13.84)	60.74 (5.49)
	Relative (k)	56.00 (0.22)	52.29 (1.23)	23.00 (5.06)	95.70 (3.57)	79.66 (1.94)	26.08 (17.72)	55.67 (8.05)
	Relative (k/q)	55.45 (0.73)	51.77 (1.49)	22.67 (7.54)	90.80 (10.64)	76.50 (7.72)	39.51 (18.52)	56.83 (11.69)
Basque	Sinusoidal	68.44 (0.88)	64.15 (1.30)	48.65 (24.10)	97.01 (0.70)	70.63 (7.73)	61.65 (2.26)	70.16 (5.46)
	Absolute	69.46 (3.29)	65.36 (2.70)	45.42 (5.55)	91.64 (3.68)	69.19 (3.87)	36.91 (6.75)	58.51 (3.31)
	TUPE (a)	63.11 (0.36)	59.23 (1.35)	6.38 (2.17)	71.92 (12.96)	57.18 (7.90)	28.21 (16.17)	39.13 (10.48)
	TUPE (r)	60.82 (0.86)	56.61 (1.77)	10.85 (2.47)	77.57 (12.22)	64.37 (6.83)	33.41 (12.54)	43.95 (8.01)
	Relative (k)	60.03 (0.57)	54.70 (0.53)	9.49 (4.37)	61.06 (20.25)	56.12 (14.21)	21.28 (15.21)	36.27 (12.29)
	Relative (k/q)	57.66 (0.96)	52.51 (1.41)	11.74 (6.91)	63.13 (32.58)	55.84 (22.94)	33.70 (18.55)	41.27 (20.79)
Finnish	Sinusoidal	67.28 (1.54)	64.20 (1.88)	51.99 (23.78)	98.90 (0.25)	76.33 (5.69)	69.72 (1.44)	74.51 (5.15)
	Absolute	67.14 (2.20)	64.04 (2.76)	51.91 (11.81)	97.90 (1.13)	77.54 (0.67)	63.53 (7.04)	72.83 (1.87)
	TUPE (a)	61.53 (1.21)	58.06 (0.95)	11.45 (1.36)	91.69 (4.67)	71.60 (3.69)	34.34 (16.61)	50.91 (6.36)
	TUPE (r)	61.07 (1.22)	57.54 (1.07)	15.90 (2.15)	94.00 (3.82)	75.84 (4.08)	47.95 (20.72)	58.29 (7.04)
	Relative (k)	57.05 (1.14)	53.13 (0.43)	22.31 (3.20)	96.01 (0.70)	78.96 (0.68)	20.75 (7.37)	53.76 (0.78)
	Relative (k/q)	55.37 (0.66)	51.29 (0.27)	22.97 (0.86)	91.87 (5.67)	79.77 (0.97)	22.81 (25.52)	54.63 (9.17)
French	Sinusoidal	63.92 (0.41)	61.82 (0.87)	54.10 (22.52)	99.36 (0.13)	77.65 (4.95)	70.28 (1.22)	75.76 (4.25)
	Absolute	64.76 (1.10)	62.68 (1.37)	59.12 (13.35)	99.14 (0.37)	79.57 (0.63)	68.90 (2.75)	76.28 (1.93)
	TUPE (a)	59.94 (0.79)	58.60 (0.71)	8.67 (2.15)	90.25 (7.76)	67.86 (7.47)	36.15 (7.99)	50.87 (6.57)
	TUPE (r)	58.28 (0.78)	56.14 (1.37)	19.36 (3.62)	96.64 (2.93)	78.50 (1.72)	51.63 (18.73)	63.50 (7.99)
	Relative (k)	55.18 (0.25)	52.96 (0.44)	18.84 (2.14)	91.12 (8.32)	73.30 (7.86)	37.50 (17.77)	56.07 (8.40)
	Relative (k/q)	54.35 (0.23)	52.10 (1.02)	19.01 (5.15)	90.09 (9.29)	72.22 (7.48)	26.51 (22.43)	51.88 (10.57)
Hindi	Sinusoidal	67.46 (0.27)	65.01 (0.32)	37.94 (20.74)	88.30 (4.64)	63.26 (13.46)	47.88 (6.96)	60.76 (4.85)
	Absolute	63.75 (1.05)	61.38 (0.97)	47.98 (0.66)	95.58 (2.19)	76.18 (1.95)	56.57 (12.48)	67.28 (4.54)
	TUPE (a)	61.70 (1.10)	59.57 (0.48)	6.34 (1.56)	74.79 (20.15)	58.50 (12.29)	28.73 (6.46)	40.11 (9.49)
	TUPE (r)	59.57 (1.24)	57.64 (1.75)	12.67 (6.57)	72.64 (24.01)	61.35 (17.78)	23.49 (10.80)	39.66 (14.67)
	Relative (k)	55.50 (0.60)	52.81 (0.87)	19.72 (4.83)	90.11 (9.87)	74.06 (4.06)	26.71 (13.65)	52.39 (9.33)
	Relative (k/q)	54.51 (0.21)	52.04 (0.74)	23.45 (6.88)	93.27 (6.35)	75.60 (3.41)	32.95 (21.46)	56.72 (10.94)
Swedish	Sinusoidal	64.62 (0.86)	62.32 (0.73)	57.11 (21.45)	99.32 (0.24)	79.89 (2.06)	71.82 (7.63)	77.39 (5.05)
	Absolute	65.54 (1.86)	62.95 (2.48)	53.52 (4.43)	97.45 (0.79)	78.85 (1.69)	66.33 (3.09)	74.58 (2.28)
	TUPE (a)	60.16 (0.99)	58.23 (1.49)	14.47 (3.45)	96.36 (2.49)	76.62 (3.05)	48.12 (19.76)	60.16 (7.98)
	TUPE (r)	58.29 (0.17)	56.12 (0.72)	21.57 (3.06)	97.80 (1.40)	79.05 (3.18)	54.56 (2.06)	64.86 (1.83)
	Relative (k)	54.94 (0.61)	52.28 (0.68)	28.17 (3.30)	98.52 (0.68)	82.81 (1.08)	56.60 (9.17)	68.09 (5.04)
	Relative (k/q)	53.66 (0.22)	51.16 (1.11)	27.87 (4.86)	97.63 (2.04)	82.11 (2.22)	49.65 (17.66)	65.89 (8.57)

Bible	Embedding	Perplexity		Retrieval		Translation		ML score
		Full	L1	0	8	0	8	
Arabic	Sinusoidal	228.44 (5.29)	193.76 (3.88)	2.30 (0.35)	32.79 (4.29)	24.54 (5.24)	13.20 (3.62)	18.21 (3.09)
	Absolute	236.83 (8.40)	206.44 (10.04)	4.18 (0.20)	22.72 (5.92)	24.18 (3.04)	5.75 (1.87)	14.21 (2.67)
	TUPE (a)	251.32 (5.55)	209.54 (9.00)	0.72 (0.36)	10.18 (6.12)	9.91 (5.73)	3.44 (2.47)	6.07 (3.67)
	TUPE (r)	228.09 (18.36)	193.67 (7.37)	1.30 (0.22)	21.49 (6.68)	23.25 (3.57)	4.19 (1.11)	12.56 (2.28)
	Relative (k)	206.77 (4.36)	177.43 (3.93)	0.84 (0.08)	8.79 (1.82)	11.85 (1.24)	3.24 (0.64)	6.18 (0.73)
	Relative (k/q)	195.47 (5.84)	166.20 (2.53)	1.44 (0.60)	15.87 (8.02)	19.30 (7.44)	4.82 (2.64)	10.36 (4.52)
German	Sinusoidal	214.19 (8.29)	185.11 (3.92)	6.80 (1.86)	76.04 (3.25)	64.60 (2.83)	37.76 (5.90)	46.30 (2.53)
	Absolute	230.05 (7.75)	205.46 (12.23)	9.81 (2.28)	63.69 (12.37)	51.91 (5.98)	17.89 (5.89)	35.83 (6.55)
	TUPE (a)	230.92 (6.27)	203.84 (2.26)	1.75 (0.39)	33.91 (2.86)	36.33 (6.21)	9.59 (2.96)	20.40 (2.65)
	TUPE (r)	211.44 (6.19)	196.39 (7.09)	2.52 (0.56)	48.23 (20.03)	46.61 (14.36)	14.43 (6.78)	27.95 (9.95)
	Relative (k)	189.45 (4.10)	165.36 (2.59)	1.50 (0.48)	16.44 (7.67)	19.43 (8.06)	5.12 (2.08)	10.62 (4.56)
	Relative (k/q)	191.06 (0.27)	168.70 (6.74)	2.33 (0.67)	33.21 (15.24)	36.76 (13.08)	11.69 (6.65)	21.00 (8.84)
Basque	Sinusoidal	236.91 (9.77)	197.71 (11.45)	1.47 (0.37)	15.82 (2.59)	13.57 (3.44)	6.47 (2.72)	9.33 (2.27)
	Absolute	248.87 (18.53)	212.26 (14.12)	4.90 (0.82)	24.07 (4.75)	17.72 (2.05)	6.00 (2.44)	13.17 (1.15)
	TUPE (a)	265.28 (11.20)	220.59 (17.75)	0.42 (0.18)	2.98 (1.30)	4.57 (1.17)	1.21 (0.57)	2.29 (0.79)
	TUPE (r)	239.52 (8.88)	196.16 (12.43)	1.04 (0.21)	11.23 (1.54)	14.64 (4.25)	3.29 (0.98)	7.55 (1.36)
	Relative (k)	214.59 (5.79)	170.40 (2.68)	0.52 (0.14)	3.24 (1.20)	6.23 (1.05)	1.60 (0.67)	2.90 (0.62)
	Relative (k/q)	206.33 (6.43)	166.16 (2.67)	0.49 (0.18)	3.15 (1.43)	5.96 (2.18)	1.37 (0.86)	2.74 (1.09)
Finnish	Sinusoidal	215.28 (2.80)	181.92 (5.08)	4.34 (0.29)	64.24 (9.50)	56.92 (5.55)	28.02 (8.05)	38.38 (5.73)
	Absolute	230.22 (12.83)	194.61 (4.17)	14.40 (3.69)	68.82 (5.06)	63.92 (5.37)	26.18 (8.15)	43.33 (5.03)
	TUPE (a)	251.91 (4.75)	215.41 (10.58)	1.94 (0.56)	35.29 (11.75)	36.96 (8.56)	10.58 (6.28)	21.19 (6.73)
	TUPE (r)	240.35 (7.43)	206.09 (14.62)	3.00 (0.40)	52.44 (7.90)	53.54 (5.29)	9.22 (5.12)	29.55 (3.15)
	Relative (k)	202.40 (8.09)	170.90 (9.54)	1.77 (0.90)	23.66 (13.30)	30.67 (14.17)	11.30 (5.45)	16.85 (8.23)
	Relative (k/q)	197.92 (5.13)	161.33 (8.55)	2.38 (0.70)	29.05 (3.31)	40.63 (10.06)	12.74 (5.02)	21.20 (4.70)
French	Sinusoidal	202.44 (4.97)	179.19 (6.13)	7.36 (1.66)	82.96 (1.61)	74.46 (1.00)	44.95 (4.13)	52.43 (0.76)
	Absolute	217.10 (11.25)	199.77 (10.51)	13.60 (1.64)	75.18 (7.83)	74.33 (5.80)	35.35 (9.98)	49.62 (6.21)
	TUPE (a)	220.48 (9.85)	203.21 (12.12)	2.49 (0.35)	52.95 (15.51)	46.47 (10.33)	15.98 (5.82)	29.47 (7.80)
	TUPE (r)	204.57 (8.86)	188.30 (14.18)	4.07 (0.72)	71.33 (10.07)	65.27 (6.85)	29.41 (8.71)	42.52 (6.50)
	Relative (k)	181.31 (2.67)	161.53 (3.44)	2.36 (1.09)	36.83 (20.95)	33.84 (15.39)	12.17 (9.58)	21.30 (10.92)
	Relative (k/q)	179.43 (9.50)	159.12 (8.66)	4.01 (0.93)	57.36 (10.80)	56.62 (9.44)	24.81 (10.53)	35.70 (6.91)
Hindi	Sinusoidal	207.71 (3.54)	187.60 (4.39)	2.08 (0.53)	22.03 (8.05)	17.30 (5.85)	7.54 (2.13)	12.24 (4.13)
	Absolute	221.81 (6.29)	197.47 (9.65)	4.62 (1.64)	29.81 (13.95)	23.44 (9.49)	6.00 (1.86)	15.97 (6.55)
	TUPE (a)	235.64 (11.94)	216.64 (15.13)	0.70 (0.06)	5.63 (1.00)	9.15 (2.07)	2.31 (0.57)	4.45 (0.69)
	TUPE (r)	210.78 (8.59)	189.10 (9.12)	0.83 (0.37)	12.64 (5.58)	13.41 (6.40)	3.26 (1.76)	7.54 (3.52)
	Relative (k)	190.88 (4.72)	172.32 (7.12)	1.24 (0.43)	15.41 (6.60)	14.80 (6.91)	4.71 (2.33)	9.04 (4.06)
	Relative (k/q)	190.68 (6.60)	168.92 (4.95)	1.06 (0.32)	11.42 (5.32)	13.01 (4.36)	3.59 (0.56)	7.27 (2.62)
Swedish	Sinusoidal	205.42 (4.67)	180.45 (2.27)	9.41 (0.30)	84.72 (2.59)	78.21 (0.52)	55.40 (6.57)	56.94 (2.23)
	Absolute	220.06 (10.32)	197.17 (5.13)	15.82 (0.56)	83.29 (0.69)	77.68 (1.00)	38.35 (4.80)	53.79 (1.34)
	TUPE (a)	220.78 (8.49)	201.86 (6.14)	3.56 (0.64)	61.11 (12.33)	58.59 (11.18)	13.75 (3.21)	34.25 (6.03)
	TUPE (r)	212.54 (3.02)	192.31 (9.32)	5.00 (1.46)	72.05 (11.82)	66.01 (9.39)	33.61 (10.73)	44.17 (7.93)
	Relative (k)	184.79 (6.02)	165.70 (6.85)	5.56 (1.40)	72.94 (3.63)	69.32 (5.43)	14.18 (5.07)	40.50 (2.83)
	Relative (k/q)	177.74 (6.85)	160.42 (5.99)	5.08 (0.24)	71.11 (5.47)	67.52 (3.27)	32.22 (12.51)	43.99 (1.05)

Table VIII.5: Full results (mean/std. over three seeds)

Paper IX

Word Order Does Matter And Shuffled Language Models Know It

***Mostafa Abdou, *Vinit Ravishankar, Artur Kulmizev, Anders Søgaard**

Appears in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, May 2022, pp. 6907–6919. DOI: 10.18653/v1/2022.acl-long.476.

Abstract

Recent studies have shown that language models pretrained and/or fine-tuned on randomly permuted sentences exhibit competitive performance on GLUE, putting into question the importance of word order information. Somewhat counter-intuitively, some of these studies also report that position embeddings appear to be crucial for models' good performance with shuffled text. We probe these language models for word order information and investigate what position embeddings learned from shuffled text encode, showing that these models retain information pertaining to the original, naturalistic word order. We show this is in part due to a subtlety in how shuffling is implemented in previous work – *before* rather than *after* subword segmentation. Surprisingly, we find even Language models trained on text shuffled *after* subword segmentation retain some semblance of information about word order because of the statistical dependencies between sentence length and unigram probabilities. Finally, we show that beyond GLUE, a variety of language understanding tasks do require word order information, often to an extent that cannot be learned through fine-tuning.

Contents

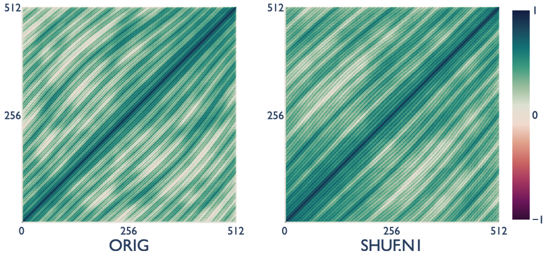


Figure IX.1: Pearson correlations between position embeddings for full-scale models; the patterns are similar to fully learnable absolute embeddings (Wang et al., 2021) and can be said to have learned something about position. We later demonstrate that this is not the case with post-BPE scrambling.

IX.1 Introduction

Transformers Vaswani et al., 2017, when used in the context of masked language modelling Devlin et al., 2019, consume their inputs concurrently. There is no notion of inherent order, unlike in autoregressive setups, where the input is consumed token by token. To compensate for this absence of linear order, the transformer architecture originally proposed in Vaswani et al. (2017) includes a fixed, sinusoidal position embedding added to each token embedding; each token carries a different position embedding, corresponding to its position in the sentence. The transformer-based BERT (Devlin et al., 2019) replaces these fixed sinusoidal embeddings with unique, learned embeddings per position; RoBERTa (Liu et al., 2019), the model investigated in this work, does the same.

Position embeddings are the only source of order information in these models; in their absence, contextual representations generated for tokens are independent of the actual position of the tokens in a sentence, and the models thus resemble heavily overparameterised bags-of-words. Sinha et al. (2021a) pre-trained RoBERTa models on shuffled corpora to demonstrate that the performance gap between these 'shuffled' language models and models trained on unshuffled corpora is minor (when fine-tuned and evaluated downstream on the GLUE (Wang et al., 2018) benchmark). They further show that this gap is considerably wider when a model is pre-trained without position embeddings. In this paper, we attempt to shed some light on why these models behave the way they do, and in doing so, seek to answer a set of pertinent questions:

- Do shuffled language models still have traces of word order information?

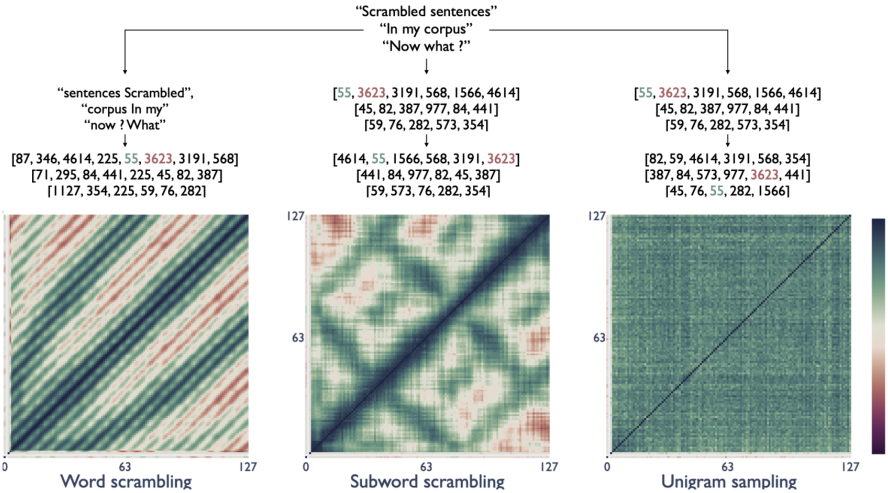


Figure IX.2: Correlations between position embeddings when shuffling training data *before* segmentation (left), i.e, at the word level, and *after* segmentation (middle), i.e., at the subword level, as well as when replacing all subwords with random subwords based on their corpus-level frequencies (right). The latter removes any dependency between subword probability and sentence length. The plots show that shuffling before segmentation retains more order information than shuffling after, and that even when shuffling after segmentation, position embeddings are meaningful because of the dependence between subword probability and sentence length.

- Why is there a gap in performance between models *without* position embeddings and models trained on shuffled tokens, with the latter performing better?
- Are there NLU benchmarks, other than GLUE, on which shuffled language models perform poorly?

Contributions We first demonstrate, in Section IX.3, that shuffled language models *do* contain word order information, and are quite responsive to simple tests for word order information, particularly when compared to models trained without position representations. In Section IX.4, we demonstrate that pre-training is sufficient to learn this: position embeddings provide the appropriate inductive bias, and performing BPE segmentation after shuffling results in sensible n-grams appearing in the pre-training corpus; this gives models the capacity to learn word order within smaller local windows. Other minor cues - like correlations between sentence lengths and token distributions - also play a role. We further corroborate our analysis by examining attention patterns across models in Sec. IX.5. In Section IX.6, we show that, while shuffled models might be almost as good as their un-shuffled counterparts on GLUE tasks, there exist NLU benchmarks that do require word order information to an extent that cannot be learned through fine-tuning alone. Finally, in Section IX.7, we describe miscellaneous experiments addressing the utility of positional embeddings when added just prior to fine-tuning.

IX.2 Models

Sinha et al. (2021a) train several full-scale RoBERTa language models on the Toronto Book Corpus (Zhu et al., 2015) and English Wikipedia.[1] Four of their models are trained on shuffled text, i.e., sentences in which n-grams are reordered at random.[2] We dub the original, unperturbed model ORIG, and the scrambled models SHUF.N1, SHUF.N2, SHUF.N3 and SHUF.N4 depending on the size of the shuffled n-grams: SHUF.N1 reorders the unigrams in a sentence, SHUF.N2 reorders its bigrams, etc. For comparison, Sinha et al. (2021a) also train a RoBERTa language model entirely *without* position embeddings (NOPOS), as well as a RoBERTa language model trained on a corpus drawn solely from unigram distributions of the original Book Corpus, i.e., a reshuffling of the entire corpus (SHUF.CORPUS). We experiment with their models, as well as with smaller models that we can train with a smaller carbon footprint. To this end, we downscale the RoBERTa architecture used in Sinha et al. (2021a). Concretely, we train single-headed RoBERTa models, dividing the embedding and feed-forward dimensionality by 12, for 24 hours on a single GPU, on 100k sentences sampled from the Toronto Book Corpus. To this end, we train a custom vocabulary of size 5,000, which we use for indexing in all our subsequent experiments. While these smaller models are in no way meant to be fine-tuned and used downstream, they are useful proofs-of-concept that we later analyse.

IX.3 Probing for word order

We begin by attempting to ascertain the extent to which shuffled language models are actually capable of encoding information pertaining to the naturalistic word order of sentences. We perform two simple tests on the full-scale models, in line with Wang and Chen (2020): the first of these is a classification task where a logistic regressor is trained to predict whether a randomly sampled token precedes another in an unshuffled sentence, and the second involves predicting the position of a word in an unshuffled sentence. The fact that we *do not* fine-tune any of the model parameters is noteworthy: the linear models can only learn word order information if it reflects in the representations the models generate somehow.

Pairwise Classification For this experiment, we train a logistic regression classification model on word representations extracted from the final layer of

[1] Training reportedly takes 72 hours on 64 GPUs.

[2] The shuffling procedure does not reorder tokens *completely* at random, but moves a token in position i to a *new* position selected at random among positions $j \neq i$.

Model	Classification (acc.)			Regression (R^2)
	2k	5k	10k	-
ORIG	81.50	81.74	80.40	0.68
SHUF.N1	65.96	64.98	71.82	0.60
NOPOS	50.41	53.35	50.22	0.03

Table IX.1: Pairwise classification and regression results.

the Transformer encoder, mean pooling over sub-tokens when required. For each word pair x and y, the classifier is given a concatenation of our model m's induced representations $m(x) \oplus m(y)$ and trained to predict a label indicating whether x precedes y or not. Holding out two randomly sampled positions, we use a training sets sized 2k, 5k, and 10k, from the Universal Dependencies English-GUM corpus (Zeldes, 2017) (excluding sentences with more than 30 tokens to increase learnability) and a test set of size $2,000$. We report the mean accuracy from three runs.

Regression Using the same data, we also train a ridge-regularised linear regression model to predict the position of a word $p(x)$ in an unshuffled sentence, given that word's model-induced representation $m(x)$. R^2 score is reported per model. To prevent the regressors from memorising word to position mappings, we perform 6-fold cross-validation, where the heldout part of the data contains no vocabulary overlap with the corresponding train set.

Results For both tasks (see Table IX.1), our results indicate that position encodings are particularly important for encoding word order: Classifiers and regressors trained on representations from ORIG and SHUF.N1 achieve high accuracies and R^2 scores, while those for NOPOS are close to random. Both ORIG and SHUF.N1 appear to be better than random given only 2k examples. These results imply that, given positional encodings and a modest training set of 2k or more examples, a simple linear model is capable of extracting word order information, enabling almost perfect extrapolation to unseen positions. Whether the position encodings come from a model trained on natural or shuffled text does not appear to matter, emphasizing that shuffled language models do indeed contain substantial information about the original word order.

IX.4 Hidden word-order signals

In Section IX.3, we observed that Sinha et al. (2021a)'s shuffled language models surprisingly exhibit information about naturalistic word order. That these models contain positional information can also be seen by visualizing position embedding similarity. Figure IX.1 displays Pearson correlations[3] for position embeddings with themselves, across positions. Here, we see that the shuffled models satisfy the idealised criteria for position embeddings described by Wang et al. (2021): namely, they appear to be a) monotonous within smaller context windows, and b) invariant to translation. If position embedding correlations are consistent across offsets over the entire space of embeddings, the model can be said to have 'learned' distances between tokens. Since transformers process all positions in parallel, and since language models without position embeddings do not exhibit such information, position embeddings have to be the source of this information. In what follows, we discuss this apparent paradox.

Subword vs. word shuffling An important detail when running experiments on shuffled text, is *when* the shuffling operation takes place. When tokens are shuffled *before* BPE segmentation, this leads to word-level shuffling, in which sequences of subwords that form words remain contiguous. Such sequences become a consistent, meaningful signal for language modelling, allowing models to efficiently utilise the inductive bias provided by position embeddings. Thus, even though our pretrained models have, in theory, not seen consecutive tokens in their pre-training data, they have learned to utilise positional embeddings to pay attention to adjacent tokens. The influence of this is somewhat visible in Figure IX.2: while models trained on text shuffled before and after segmentation both exhibit shifts in the *polarity* of their position correlations, only the former show bands of varying *magnitude*, similar to the full-scale models. Ravishankar and Søgaard (2021) discuss the implications of these patterns in a multilingual context; we hypothesise that in our context, the periodicity in magnitude is a visible artefact of the model's ability to leverage position embeddings to enable offset attention. In Section IX.5, we analyse the effect of shuffling the pre-training data on the models' attention mechanisms.

Accidental overlap In addition to the n-gram information which results from shuffling before segmentation, we also note that short sentences tend to include original bigrams with high probability, leading to stronger associations for words that are adjacent in the original texts. This effect is obviously much stronger when shuffling before segmentation than after segmentation. Figure IX.3 shows how frequent overlapping bigrams (of any sort) are, comparing word and subword shuffling over 50k sentences.

Sentence length Finally, we observe some preserved information about the original word order even when shuffling is performed *after* segmentation. We hypothesize that this is a side-effect of the non-random relationship between sentence length and unigram probabilities. That unigram probabilities correlate with sentence length follows from the fact that different genres exhibit different sentence length distributions Jin and Liu, 2017; Sigurd et al., 2004. Also, some

[3] We see similar patterns with dot products for all our plots; we use Pearson correlations to constrain our range to $[-1, 1]$.

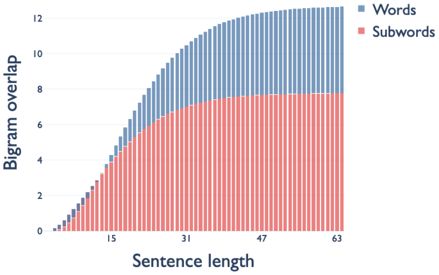


Figure IX.3: (Cumulative) plot showing subword bigram overlap after shuffling either words or subwords, as a percentage of the total number of seen bigrams. We see the overlap is significant, especially when performing shuffling before segmentation.

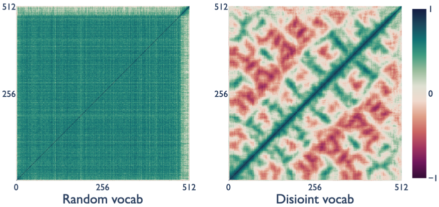


Figure IX.4: Similarity matrix between models with sentences sampled based on unigram corpus statistics; disjoint vocab implies a correlation between token choice and sentence length.

words occur very frequently in formulaic contexts, e.g., *thank* in *thank you*. This potentially means that there is an approximately learnable relationship between the distribution of words and sentence boundary symbols.

To test for this, we train two smaller language models on unigram-sampled corpora: for the first, we use the first 100k BookCorpus sentences as our corpus, shuffling tokens at a corpus level (yet keeping the original sentence lengths). The stark difference in position embedding correlations between that and shuffling is seen in Figure IX.2. For the second, we sample from two different unigram distributions: one for short sentences and one for longer sentences (details in Appendix IX.B). While the first model induces no correlations at all, the second does, as shown in Figure IX.4, implying that sentence length and unigram occurrences is enough to learn *some* order information.

IX.5 Attention analysis

Transformer-based language models commonly have attention heads that attend to neighboring positions **voita-etal-2019-analyzing**; Ravishankar et al., 2021.

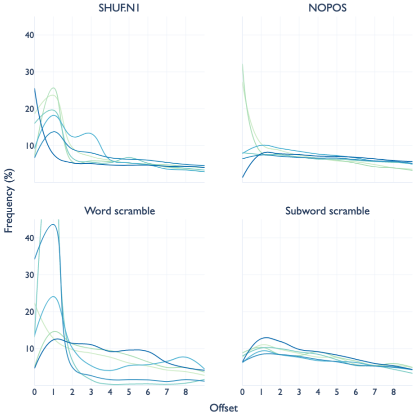


Figure IX.5: Relative frequency of offsets between token pairs in an attention relation; the y-axis denotes the percentage of total attention relations that occur at the offset indicated on the x-axis. We plot layers $l \in \{1, 2, 7, 8, 11, 12\}$ with increasing line darkness.

Such attention heads are positional and can only be learned in the presence of order information. We attempt to visualise the attention mechanism for pre-trained models by calculating, for each head and layer, the offset between a token and the token that it pays maximum attention to[4]. We then plot how frequent each offset is, as a percentage, over 100 Book Corpus sentences, in Figure IX.5, where we present results for two full-scale models, and two smaller models (see §2). When compared to Nopos, Shuf.n1 has a less uniform pattern to its attention mechanism: it is likely, even at layer 0, to prefer to pay attention to adjacent tokens, somewhat mimicking a convolutional window (Cordonnier et al., 2020). We see very similar differences in distribution between our smaller models: Shuffling after segmentation, i.e., at the subword level, influences early attention patterns.

IX.6 Evaluation beyond GLUE

SuperGLUE and WinoGrande Sinha et al. (2021a)'s investigation is conducted on GLUE and on the Paraphrase Adversaries from Word shuffling (PAWS) dataset Zhang et al., 2019. For these datasets, they find that models pretrained on shuffled text perform only marginally worse than those pretrained on normal text. This result, they argue can be explained in two ways: either a) these tasks do not need word order information to be solved, or b) the required word order information can be acquired during finetuning. While GLUE has been a useful

[4] This method of visualisation is somewhat limited, in that it examines only the *maximum* attention paid by each token. We provide more detailed plots over attention *distributions* in the Appendix.

benchmark, several of the tasks which constitute it have been shown to be solvable using various spurious artefacts and heuristics Gururangan et al., 2018; Poliak et al., 2018. If, for instance, through finetuning, models are learning to rely on such heuristics as lexical overlap for MNLI McCoy et al., 2019, then it is unsurprising that their performance is not greatly impacted by the lack of word order information.

Evaluating on the more rigorous set of SuperGLUE tasks[5] Wang et al., 2019 and on the adversarially-filtered Winograd Schema examples Levesque et al., 2012 of the WinoGrande dataset Sakaguchi et al., 2020 produces results which paint a more nuanced picture compared to those of Sinha et al. (2021a). The results, presented in Table IX.2, show accuracy or F1 scores for all models. For two of the tasks (MultiRC Khashabi et al., 2018, COPA Roemmele et al., 2011), we observe a pattern in line with that seen in Sinha et al. (2021a)'s GLUE and PAWS results: the drop in performance from ORIG to SHUF.N1 is minimal (mean: 1.75 points; mean across GLUE tasks: 3.3 points)[6], while that to NOPOS is more substantial (mean: 10.5 points; mean across GLUE tasks: 18.6 points).

This pattern alters for the BoolQ Yes/No question answering dataset Clark et al., 2019, the CommitmentBank De Marneffe et al., 2019, the ReCoRD reading comprehension dataset Zhang et al., 2018, both the Winograd Schema tasks, and to some extent the Words in Context dataset Pilehvar and Camacho-Collados, 2018. For these tasks we observe a larger gap between ORIG and SHUF.N1 (mean: 8.1 points), and an even larger one between ORIG and NOPOS (mean: 19.78 points). We note that this latter set of tasks requires inferences which are more context-sensitive, in comparison to the two other tasks or to the GLUE tasks.

Consider the Winograd schema tasks, for example. Each instance takes the form of a binary test with a statement comprising of two possible referents (blue) and a pronoun (red) such as: `Sid explained his theory to Mark but he couldn't convince him.` The correct referent of the pronoun must be inferred based on a special discriminatory segment (underlined). In the above example, this depends on a) the identification of "Sid" as the subject of "explained" and b) inferring that the pronoun serving as the subject of "convinced" should refer to the same entity. Since the Winograd schema examples are designed so that the referents are equally associated with their context[7], word order is crucial[8] for establishing the roles of "Sid" and "Mark" as subject and object of "explained" and "he" and "him" as those of "convinced". If these roles cannot be established, making the correct inference becomes impossible.

A similar reasoning can be applied to the Words in Context dataset and the

[5] Results are reported for an average of 3 runs per task. The RTE task is excluded from our results as it is also part of GLUE; RTE results can be found in Sinha et al. (2021a).

[6] CoLA results are excluded from the GLUE calculations due to the very high variance across random seeds reported by Sinha et al. (2021a).

[7] e.g. Sid and Mark are both equally likely subjects/objects here. Not all Winograd schema examples are perfect in this regard, however, which could explain why scrambled models still perform above random. See Trichelair et al. (2018) for a discussion of the latter point.

[8] Particularly in a language with limited morphological role marking such as English.

Model	BoolQ	CB	COPA	MultiRC	ReCoRD	WiC	WSC	WinoGrande
ORIG	77.6	88.2 / 87.4	61.6	67.8 / 21.9	73.5 / 72.8	67.4	73.5	62.9
SHUF.N1	72.4	79.7 / 82.5	59.7	66.2 / 15.0	61.1 / 60.4	63.0	62.9	55.7
SHUF.N2	73.1	86.6 / 85.5	60.3	64.8 / 16.1	63.1 / 62.4	63.0	65.3	57.6
SHUF.N4	73.5	87.9 / 87.1	60.8	66.2 / 18.2	64.6 / 63.9	62.4	65.3	59.53
NOPOS	66.0	63.5 / 75.0	55.6	52.8 / 3.8	23.8 / 23.5	55.4	63.09	52.73
SHUF.CORPUS	66.7	65.6 / 73.8	56.1	52.6 / 6.4	31.0 / 30.3	57.3	65.14	51.68

Table IX.2: SuperGLUE and WinoGrande results for all models. Scores displayed are: Avg. F1 / Accuracy for CB; F1a / Exact Match for MultiRC; F1 / Accuracy for ReCoRD ; accuracy for the remaining tasks.

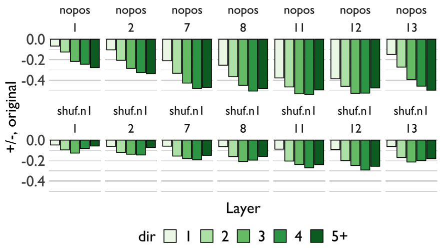


Figure IX.6: Δ, dependency arcs probing accuracy across lengths 1-5+, w.r.t. ORIG.

CommitmentBank. The former task tests the ability of a model to distinguish the senses of a polysemous word based on context. While this might often be feasible via a notion of contextual association that higher-order distributional statistics are sufficient for, some instances will require awareness of the word's role as an argument in the sentence. The latter task investigates the projectivity of finite clausal complements under entailment cancelling operators. This is dependent on both the scope of the entailment operator and the identity of the subject of the matrix predicate De Marneffe et al., 2019, both of which are sensitive to word order information.

A final consideration to take into account is dataset filtering. Two of the tasks where we observe the largest difference between ORIG, SHUF.N1, and NOPOS — WinoGrande and ReCoRD — apply filtering algorithms to remove cues or biases which would enable models to heuristically solve the tasks. This indicates that by filtering out examples containing cues that make them solvable via higher order statistics, such filtering strategies do succeed at compelling models to (at least partially) rely on word order information.

Dependency Tree Probing Besides GLUE and PAWS, Sinha et al. (2021a)'s analysis also includes several probing experiments, wherein they attempt to decode dependency tree structure from model representations. They show, interestingly, that the SHUF.N4, SHUF.N3 and SHUF.N2 models perform only marginally worse than ORIG, with SHUF.N1 producing the lowest scores (lower, in fact, than SHUF.CORPUS). Given the findings of Section IX.3, we are interested in taking a closer look at this phenomenon. Here, we surmise that *dependency length* plays a crucial role in the probing setup, where permuted models may succeed on par with ORIG in capturing local, adjacent dependencies, but increasingly struggle to decode longer ones. To evaluate the extent to which this is true, we train a bilinear probe (used in Hewitt and Liang (2019)) on top of all model representations and evaluate its accuracy across dependencies binned by length, where length between words w_i and w_j is defined as $|i - j|$. We opt for using the bilinear probe over the Pareto probing framework (Pimentel et al., 2020), as the former learns a transformation directly over model representations, while the latter adds the parent and child MLP units from Dozat et al. (2017) – acting more like a parser. We train probes on the English Web Treebank (Silveira et al., 2014) and evaluate using UAS, the standard parsing metric.

Figure IX.6 shows Δ probing accuracy across various dependency lengths for NOPOS and SHUF.N1, with respect to ORIG[9]; we include detailed Δs for all models in Appendix IX.C. For NOPOS, parsing difficulty increases almost linearly with distance, often mimicking the actual frequency distribution of dependencies at these distances in the original treebank (Appendix IX.C); for SHUF.N1, the picture is a lot more nuanced, with dependencies at a distance of 1 consistently being closer in terms of parseability to ORIG, which, we hypothesise, is due to its adjacency bias.

IX.7 Other Findings

Random position embeddings are difficult to add post-training We tried to quantify the degree to which the inductive bias imparted by positional embeddings can be utilised, solely via fine-tuning. To do so, for a subset of GLUE tasks (MNLI, QNLI, RTE, SST-2, CoLA), we evaluate NOPOS, and a variant where we randomly initialised learnable position embeddings and add them to the model, with the rest of the model equivalent to NOPOS. We see no improvement in results, except for MNLI, that we hypothesise stems from position embeddings acting as some sort of regularisation parameter. To test this, we repeat the above set of experiments, this time injecting Gaussian noise instead; this has been empirically shown to have a regularising effect on the network (Bishop, 1995; Camuto et al., 2021). Adding Gaussian noise led to a slight increase in score for just MNLI, backing up our regularisation hypothesis.

Models learn to expect specific embeddings Replacing the positional embeddings in ORIG with fixed, sinusoidal embeddings before fine-tuning significantly hurts scores on the same subset of GLUE tasks, implying that the models expect embeddings that resemble the inductive bias imparted by random embeddings, and that fine-tuning tasks do not have sufficient data to overcome this. The addition of fixed, *sinusoidal* to NOPOS also does not

[9] Note that Layer 13 refers to a linear mix of all model layers, as is done for ELMo (Peters et al., 2018).

improve model performance on a similar subset of tasks; this implies, given that sinusoidal embeddings are already meaningful, that model weights also need to learn to fit the embeddings they are given, and that they need a substantial amount of data to do so.

IX.8 On Word Order

In Humans It is generally accepted that a majority of languages have "canonical" or "base' word orderings Comrie, 1989 (e.g. Subject-Verb-Object in English, and Subject-Object-Verb in Hindi). Linguists consider word order to be a *coding property* — mechanisms by which abstract, syntactic structure is encoded in the surface form of utterances. Beyond word order, other coding properties include, e.g. subject-verb agreement, morphological case marking, or function words such as adpositions. In English, word order is among the most prominent coding properties, playing a crucial role in the expression of the main verb's core arguments: subject and object. For more morphologically complex languages, on the other hand, (e.g. Finnish and Turkish), word order is primarily used to convey pragmatic information such as topicalisation or focus. In such cases, argument structure is often signalled via case-marking, where numerous orderings become possible (shift in topic or focus nonwithstanding). We refer the reader to Kulmizev and Nivre (2021) for a broader discussion of these topics and their implications when studying syntax through language models.

More generally, evidence for the saliency of word order in linguistic processing and comprehension comes from a variety of studies using acceptability judgements, eye-tracking data, and neural response measurements Bahlmann et al., 2007; Bever, 1970; Danks and Glucksberg, 1971; Ding et al., 2016; Fedorenko et al., 2016; Friederici et al., 2000; Friederici et al., 2001; Just and Carpenter, 1980; Lerner et al., 2011; Pallier et al., 2011. Psycholinguistic research has, however, also highlighted the robustness of sentence processing mechanisms to a variety of perturbations, including those which violate word order restrictions Ferreira et al., 2002; Gibson et al., 2013; Traxler, 2014. In recent work, Mollica et al. (2020) tested the hypothesis that composition is the core function of the brain's language-selective network and that it can take place even when grammatical word order constrains are violated. Their findings confirmed this, showing that stimuli with shuffled word order where local dependencies were preserved — as is, roughly speaking, the case for many dependencies in the sentences SHUF.N4 is trained on — elicited a neural response in the language network that is comparable to that elicited by normal sentences. When interword dependencies were disrupted so combinable words were so far apart that composition among nearby words was highly unlikely — as in SHUF.N1, neural response fell to a level compared to unconnected word lists.

In Machines Recently, many NLP researchers have attempted to investigate the role of word order information in language models. For example, Lin et al. (2019) employ diagnostic classifiers and attention analyses to demonstrate that lower (but not higher) layers of BERT encode word order information. Papadimitriou et al. (2021) find that Multilingual BERT is sensitive to morphosyntactic alignment, where numerous languages (out of 24 total) rely on word order to mark subjecthood (English among them). Alleman et al. (2021) implement an input perturbation framework (n-gram shuffling, phrase swaps, etc.), and employ it towards testing the sensitivity of BERT's representations to various types of structure in sentences. They report a sensitivity to larger constituent units of sentences in higher layers, which they deduce to be influenced by hierarchical phrase structure. O'Connor and Andreas (2021) examine the contribution of various contextual features to the ability of GPT-2 (Radford et al., 2019) to predict upcoming tokens. Their findings show that several destructive manipulations, including in-sentence word shuffling, applied to mid- and long range contexts lead only to a modest increase in *usable information* as defined according to the V-information framework of Xu et al. (2020).

Similarly, word order information has been found not to be essential for various NLU tasks and datasets. Early work showed that Natural Language Inference tasks are largely insensitive to permutations of word order Parikh et al., 2016; Sinha et al., 2021b. Pham et al. (2020) and Gupta et al. (2021) discuss this in greater detail, demonstrating that test-time word order perturbations applied to GLUE benchmark tasks have little impact on LM performance. Following up on this, Sinha et al. (2021a), which our work builds on, found that *pretraining* on scrambled text appears to only marginally affect model performance. Most related to this study, Clouatre et al. (2022) introduce two metrics for gauging the local and global ordering of tokens in scrambled texts, observing that only the latter is altered by the perturbation functions found in prior literature. In experiments with GLUE, they find that local (sub-word) perturbations show a substantially stronger performance decay compared to global ones.

In this work, we present an in-depth analysis of these results, showing that LMs trained on scrambled text can actually retain word information and that – as for humans – their sensitivity to word order is dependent on a variety of factors such as the nature of the task and the locality of perturbation. While performance on some "understanding" evaluation tasks is not strongly affected by word order scrambling, the effect on others such as the Winograd Schema is far more evident.

IX.9 Conclusion

Much discussion has resulted from recent work showing that scrambling text at different stages of testing or training does not drastically alter the performance of language models on NLU tasks. In this work, we presented analyses painting a more nuanced picture of such findings. Primarily, we demonstrate that, as far as altered pre-training is concerned, models still do retain a semblance of word order knowledge — largely at the local level. We show that this knowledge stems from cues in the altered data, such as adjacent BPE symbols and correlations between sentence length and content. The order in which shuffling is performed — before or after BPE tokenization — is influential in models' acquisition of word order, which calls for caution in interpreting previous results. Finally, we show that there exist NLU tasks that are far more sensitive to sentence structure as expressed by word order.

Acknowledgements

We thank Stephanie Brandl, Desmond Elliott, Yova Kementchedjhieva, Douwe Kiela and Miryam de Lhoneux for their feedback and comments. We acknowledge the CSC-IT Centre for Science, Finland, for providing computational resources. Vinit worked on this paper while on a research visit to the University of Copenhagen. Mostafa and Anders are supported by a Google Focused Research Award.

References

Alleman, M. et al. (2021). "Syntactic Perturbations Reveal Representational Correlates of Hierarchical Phrase Structure in Pretrained Language Models". In: *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Online: Association for Computational Linguistics.

Bahlmann, J. et al. (2007). "An fMRI study of canonical and noncanonical word order in German". In: *Human brain mapping* vol. 28, no. 10.

Bever, T. G. (1970). "The cognitive basis for linguistic structures". In: *Cognition and the development of language*.

Bishop, C. M. (1995). "Training with Noise is Equivalent to Tikhonov Regularization". In: *Neural Computation* vol. 7, no. 1.

Camuto, A. et al. (2021). "Explicit Regularisation in Gaussian Noise Injections". In: *arXiv:2007.07368 [cs, stat]*.

Clark, C. et al. (2019). "BoolQ: Exploring the surprising difficulty of natural yes/no questions". In: *arXiv preprint arXiv:1905.10044*.

Clouatre, L. et al. (2022). "Local Structure Matters Most: Perturbation Study in NLU". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics.

Comrie, B. (1989). *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Cordonnier, J.-B. et al. (2020). "On the Relationship between Self-Attention and Convolutional Layers". In: *arXiv:1911.03584 [cs, stat]*.

Danks, J. H. and Glucksberg, S. (1971). "Psychological scaling of adjective orders". In: *Journal of Memory and Language* vol. 10, no. 1.

De Marneffe, M.-C. et al. (2019). "The CommitmentBank: Investigating projection in naturally occurring discourse". In: *proceedings of Sinn und Bedeutung*. Vol. 23. 2.

Devlin, J. et al. (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.

Ding, N. et al. (2016). "Cortical tracking of hierarchical linguistic structures in connected speech". In: *Nature neuroscience* vol. 19, no. 1.

Dozat, T. et al. (2017). "Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 Shared Task". In: *CoNLL 2017*.

Fedorenko, E. et al. (2016). "Neural correlate of the construction of sentence meaning". In: *Proceedings of the National Academy of Sciences* vol. 113, no. 41.

Ferreira, F. et al. (2002). "Good-enough representations in language comprehension". In: *Current directions in psychological science* vol. 11, no. 1.

Friederici, A. D. et al. (2000). "Auditory language comprehension: an event-related fMRI study on the processing of syntactic and lexical information". In: *Brain and language* vol. 74, no. 2.

Friederici, A. D. et al. (2001). "Syntactic parsing preferences and their on-line revisions: A spatio-temporal analysis of event-related brain potentials". In: *Cognitive Brain Research* vol. 11, no. 2.

Gibson, E. et al. (2013). "Rational integration of noisy evidence and prior semantic expectations in sentence interpretation". In: *Proceedings of the National Academy of Sciences* vol. 110, no. 20.

Gupta, A. et al. (2021). "BERT & family eat word salad: Experiments with text understanding". In: *arXiv preprint arXiv:2101.03453*.

Gururangan, S. et al. (2018). "Annotation Artifacts in Natural Language Inference Data". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics.

Hewitt, J. and Liang, P. (2019). "Designing and Interpreting Probes with Control 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.

Jin, H. and Liu, H. (2017). "How will text size influence the length of its linguistic constituents?" In: *Poznan Studies in Contemporary Linguistics* vol. 53.

Just, M. A. and Carpenter, P. A. (1980). "A theory of reading: From eye fixations to comprehension." In: *Psychological review* vol. 87, no. 4.

Khashabi, D. et al. (2018). "Looking Beyond the Surface:A Challenge Set for Reading Comprehension over Multiple Sentences". In: *NAACL*.

Kulmizev, A. and Nivre, J. (2021). "Schrödinger's Tree – On Syntax and Neural Language Models". In: *arXiv preprint arXiv:2110.08887*.

Lerner, Y. et al. (2011). "Topographic mapping of a hierarchy of temporal receptive windows using a narrated story". In: *Journal of Neuroscience* vol. 31, no. 8.

Levesque, H. et al. (2012). "The winograd schema challenge". In: *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Lin, Y. et al. (2019). "Open Sesame: Getting Inside BERT's Linguistic Knowledge". In: *arXiv preprint arXiv:1906.01698*.

Liu, Y. et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv:1907.11692 [cs]*.

McCoy, T. et al. (2019). "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". In.

Mollica, F. et al. (2020). "Composition is the core driver of the language-selective network". In: *Neurobiology of Language* vol. 1, no. 1.

O'Connor, J. and Andreas, J. (2021). "What Context Features Can Transformer Language Models Use?" In: *arXiv preprint arXiv:2106.08367*.

Pallier, C. et al. (2011). "Cortical representation of the constituent structure of sentences". In: *Proceedings of the National Academy of Sciences* vol. 108, no. 6.

Papadimitriou, I. et al. (2021). "Deep Subjecthood: Higher-Order Grammatical Features in Multilingual BERT". In: *arXiv preprint arXiv:2101.11043*.

Parikh, A. P. et al. (2016). "A decomposable attention model for natural language inference". In: *arXiv preprint arXiv:1606.01933*.

Peters, M. et al. (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.

Pham, T. M. et al. (2020). "Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks?" In: *arXiv preprint arXiv:2012.15180*.

Pilehvar, M. T. and Camacho-Collados, J. (2018). "WiC: the word-in-context dataset for evaluating context-sensitive meaning representations". In: *arXiv preprint arXiv:1808.09121*.

Pimentel, T. et al. (2020). "Pareto Probing: Trading Off Accuracy for Complexity". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Poliak, A. et al. (2018). "Hypothesis only baselines in natural language inference". In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics.

Radford, A. et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI Blog* vol. 1, no. 8.

Ravishankar, V. and Søgaard, A. (2021). "The Impact of Positional Encodings on Multilingual Compression". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Ravishankar, V. et al. (2021). "Attention Can Reflect Syntactic Structure (If You Let It)". In: *Proceedings of the 16th Conference of the European Chapter of the*

Association for Computational Linguistics: Main Volume. Online: Association for Computational Linguistics.

Roemmele, M. et al. (2011). “Choice of plausible alternatives: An evaluation of commonsense causal reasoning”. In: *2011 AAAI Spring Symposium Series*.

Sakaguchi, K. et al. (2020). “Winogrande: An adversarial winograd schema challenge at scale”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05.

Sigurd, B. et al. (2004). “Word length, sentence length and frequency – Zipf revisited”. In: *Studia Linguistica* vol. 58, no. 1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0039-3193.2004.00109.x.

Silveira, N. et al. (2014). “A Gold Standard Dependency Corpus for English.” In: *LREC*. Citeseer.

Sinha, K. et al. (2021a). “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Sinha, K. et al. (2021b). “Unnatural Language Inference”. 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.

Traxler, M. J. (2014). “Trends in syntactic parsing: Anticipation, Bayesian estimation, and good-enough parsing”. In: *Trends in cognitive sciences* vol. 18, no. 11.

Trichelair, P. et al. (2018). “On the evaluation of common-sense reasoning in natural language understanding”. In: *arXiv preprint arXiv:1811.01778*.

Vaswani, A. et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems*.

Wang, A. et al. (2018). “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics.

Wang, A. et al. (2019). “SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems”. In: *Advances in Neural Information Processing Systems*. Ed. by Wallach, H. et al. Vol. 32. Curran Associates, Inc.

Wang, Y.-A. and Chen, Y.-N. (2020). “What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Wang, B. et al. (2021). “On Position Embeddings in BERT”. In: *International Conference on Learning Representations*.

Xu, Y. et al. (2020). “A theory of usable information under computational constraints”. In: *arXiv preprint arXiv:2002.10689*.

Zeldes, A. (2017). “The GUM corpus: Creating multilayer resources in the classroom”. In: *Language Resources and Evaluation* vol. 51, no. 3.

Zhang, S. et al. (2018). “Record: Bridging the gap between human and machine commonsense reading comprehension”. In: *arXiv preprint arXiv:1810.12885*.

Zhang, Y. et al. (2019). “PAWS: Paraphrase Adversaries from Word Scrambling”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Zhu, Y. et al. (2015). “Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books”. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Appendix IX.A Subword vs. word scrambling

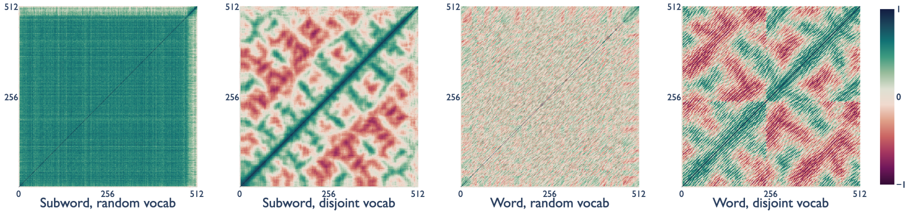


Figure IX.7: Pearson correlations, when scrambling by subword/word, with-/without disjoint vocabularies. Disjoint vocabularies appear to induce patterns in position-position correlations, while scrambling at a word level induces ‘stripes’ of oscillating magnitude; this is likely due to position embeddings learning connections to adjacent tokens.

Appendix IX.B On biased sampling

We first split our vocab of size 5,000 into two halves, both of size 2500, such that the sum total of unigram frequencies of tokens in each half is roughly equivalent. Next, iterating over 100k BookCorpus sentences, we determine the sentence length l, for which there are an equivalent number of tokens in sentences with length $< l$ and sentences with length $>= l$. We then sample tokens from the first vocab half for sentences $< l$, and from the second vocab half for sentences with length $>= l$, 80% of the time; for the other 20%, we sample from the opposite half to introduce some overlap.

Appendix IX.C Full UD results

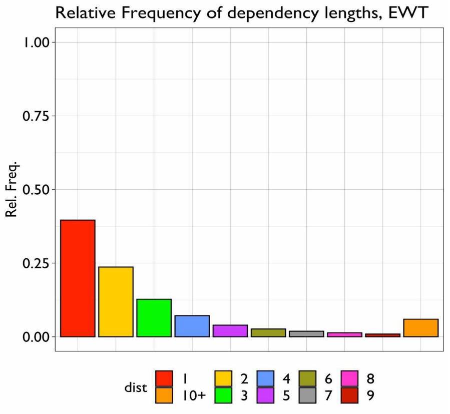


Figure IX.8: Relative frequencies of dependency relations in $UD_{English-EWT}$, at a dependency lengths indicated by the x-axis

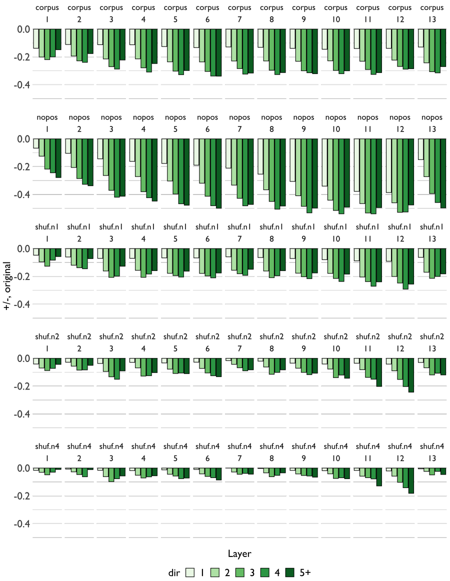


Figure IX.9: Δ UAS, all models and layers across dependency lengths 1-5+, w.r.t. ORIG. Layer 13 represents a linear mix of all model layers.