Multilingual Language Models: Analysis and Algorithms

Terra Blevins

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington
2024

Reading Committee:
Luke Zettlemoyer, Chair
Noah A. Smith
Yulia Tsvetkov

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Multilingual Language Models: Analysis and Algorithms

Terra Blevins

Chair of the Supervisory Committee:

Luke Zettlemoyer
Computer Science and Engineering

While large language models (LLMs) continue to grow in scale and gain new zero-shot capabilities, their performance for languages beyond English increasingly lags behind. This gap is due to the *curse of multilinguality*, where multilingual language models perform worse on individual languages than a monolingual model trained on that language due to inter-language competition for representation. These issues are further compounded by the disparate amounts and qualities of training data for different languages, leading to increasingly degraded performance on lower-resource languages. However, because training new large language models for individual languages is compute- and data-intensive, multilingual language models remain the de facto approach for most of the world's languages. Therefore, it remains an open question as to how we can alleviate the curse of multilinguality and build multilingual models that fairly model many languages.

This dissertation investigates how current language models do and don't capture multiple languages and examines how multilingual language models differ from monolingual ones. We first present an analysis method, *structural probing*, used for many of this work's analyses. Then, we examine the unexpected ability of monolingual language models to exhibit cross-lingual behavior, finding that this phenomenon is due to inherent language contamination of pretraining data collected

at scale. This shows that LMs can learn languages from surprisingly small subsets of their training data and implies that all language models are multilingual when trained at scale. We next characterize the pretraining dynamics of multilingual language models, showing that while multilingual models learn information about individual languages early on, cross-lingual transfer is acquired throughout the pretraining process. This analysis also demonstrates the curse of multilinguality as it develops during pretraining, causing the model to forget previously learned information.

Inspired by these insights, we propose a sparse language modeling approach for training Cross-Lingual Expert Language Models (X-ELM) to explicitly allocate parameters to different languages and reduce inter-language competition for model capacity. X-ELMs improve performance for all languages we consider, as well as provide efficiency and model adaptation benefits over prior methods. Due to these characteristics, X-ELM increases access to multilingual NLP by providing better-performing and more usable models for all languages.

Acknowledgements

My time in grad school and this dissertation would not have been possible without the help and support of my friends, family, and colleagues. Foremost, I would like to thank my advisor, Luke Zettlemoyer, from whom I've learned so much about research, writing, and how to do good science. Luke is an incredible mentor who is brilliant but also patient and kind, and I aspire to be an advisor like him.

I would also like to acknowledge:

- *my committee*, Noah Smith, Yulia Tsvetkov, Joakim Nivre, and Aylin Caliskan for their support on this work and throughout my time at UW.
- *my co-authors and collaborators*, including Hila Gonen, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Agatha Downey, Srini Iyer, Mandar Joshi, Omer Levy, Akari Asai, Shane Steinert-Threlkeld, Stephen Mayhew, Marek Šuppa, Yuval Pinter, and Weijia Shi. The very best parts of my PhD were when I was collaborating with and learning from these excellent researchers. And a particular thank you to my undergraduate and master students, with whom I have greatly enjoyed working: Shuheng Liu, Kushal Mangipudi, Alessio Tosolini, and Murray Kang.
- *my friends and lab mates in zlab and the UW NLP community* including Ari Holtzman, Julian Michael, Victor Zhong, Antoine Bosselut, Sofia Serrano, Artidoro Pagnoni, Sewon Min, Tim Dettmers, and Bhargavi Paranjape.

- *my undergraduate advisor* Kathy McKeown, and Or Biran, for taking a chance on me and getting me started in NLP research.

- *my family*, Tana, Brian, Reed, Annie, Helen, Carol, and Crookshanks, for always being there for me. This process would not have been possible without y'all. And thank you to my Tia Carla, for inspiring me to pursue research, and to my grandfathers Earl and Steve — I wish you were still here.

Last but certainly not least, thank you to my husband Jacob, for your unwavering love and steadfast support. You have been my rock throughout this journey.

DEDICATION

To Tana and Brian, for always believing in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multilingual language models (LMs) are a significant breakthrough in building language technologies for many of the world's languages. These models learn many languages simultaneously from raw text data, which has led to substantial performance improvements on many NLP tasks in these languages and the consolidation of models for different languages into a single, multilingual system. Multilingual LMs also learn to transfer information *across* languages from this text data without explicit cross-lingual supervision. This skill in multilingual models is particularly noteworthy, as it allows LMs to apply NLP tasks to languages with limited (or no) annotated data.

However, current methods for training multilingual language models come at a cost, as these LMs often perform suboptimally on individual languages within their scope. This phenomenon, called *the curse of multilinguality*, is hypothesized to occur due to limited model capacity in multilingual models when compared to similarly sized monolingual ones (Conneau et al., 2020a). As these models are often the only (large-scale) LMs available for languages other than English, this *curse* likely affects the performance of all NLP technologies for these languages that build on pretraining.

Furthermore, while all language models are affected by issues from training on large-scale text data, these problems are exacerbated in multilingual LMs due to incorporating data from many languages rather than a single one (often, English). Specific instances of these data issues for

multilingual LMs include the content and quality of large-scale text corpora (Kreutzer et al., 2022) as well as how language models learn multiple languages and cross-lingual transfer from unaligned text examples (Conneau et al., 2020b).

This dissertation investigates the issues underlying the current generation of multilingual language models. To further understand how models can learn from large-scale but noisy text data, Chapters 3 and 4 analyze the data and training dynamics of pretrained models that capture many languages. In both chapters, we use the *probing* techniques presented in Chapter 2 to analyze LMs across experimental settings. These experiments demonstrate that language models can learn information about languages from tiny portions of their training data.[1] However, these subsets are insufficient to learn languages optimally, and multilingual LMs often forget information about languages during pretraining. This provides evidence of how the curse of multilinguality develops during pretraining: languages "compete" for model capacity as they are jointly optimized with stochastic sampling, leading to some languages being overwritten throughout the training process.

Chapter 5 then proposes a new approach to multilingual language modeling (Cross-lingual Expert Language Models, or X-ELM) that resolves the curse of multilinguality by explicitly allocating model parameters to different subsets of the multilingual training data. In doing so, X-ELM both improves performance on every language considered and gains efficiency and usability benefits over current multilingual modeling approaches. Finally, Chapter 6 discusses the implications of these findings in more detail and proposes future work building on this dissertation to further understand and improve multilinguality in NLP.

1.1 Background and Related Work

This section first overviews the history and current state of multilinguality in NLP (§1.1.1), particularly regarding training and understanding multilingual language models. Then, §1.1.2 presents methods related to the technical approaches used and proposed in this dissertation.[2]

[1] Notably, even when data filtering steps are taken to remove most languages from the training corpus.

[2] Portions of this section contain material originally published in Blevins et al. (2022) and Blevins et al. (2024).

1.1.1 Overview of Multilingual Language Models

Despite their current prevalence, language models trained on more than one language were relatively understudied before pretraining.[3] Of these multilingual LMs, the majority were developed for speech processing (Uebler, 2001; Xu and Fung, 2012; Tsvetkov et al., 2016, inter alia); others addressed machine translation (e.g., Lu et al., 2012) and codeswitching (e.g., Mabokela et al., 2014) applications.

Since the rise of pretraining, multilingual LMs have become much more prevalent and are applied broadly to different language processing tasks. These studies of multilingual LMs cover different areas of focus: developing better methods for multilingual pretraining, cross-lingual transfer, and adapting multilingual LMs to new languages, as well as other lines of work on analyzing the knowledge learned by these models from the multilingual pretraining objective.

Multilingual Pretraining of Language Models The initial multilingual pretrained models (e.g., Conneau and Lample, 2019; Delvin, 2019) were proposed shortly after the rise of English pretrained models. Since then, many variations and improvements on multilingual pretraining have been introduced: by changing the architecture and scaling the model size up (Goyal et al., 2021; Lin et al., 2022), combining additional objectives to the main LM objective (Chi et al., 2022; Reid and Artetxe, 2022), careful language and data curation (Scao et al., 2022; Ogunremi et al., 2023), and scaling and balancing the vocabulary across the different languages (Liang et al., 2023). Chapter 5 presents X-ELM, which is a new method for multilingual pretraining that relaxes the assumption of using a single (*dense*) encoder model. Most similar to this approach is Pfeiffer et al. (2022), which proposes a new modular model architecture, X-MOD, that contains language-specific modules. However, many of the limitations of dense language modeling persist in this architecture since the model and modules are jointly trained.

An issue common to most methods for multilingual pretraining is the *curse of multilinguality*

[3]The trigrams "multilingual language model," "polyglot language model," and "cross-lingual language model" together occur in 112 articles on Google Scholar prior to 2018. In comparison, since the beginning of 2018 these phrases have appeared in 6,781 articles (as of May 10, 2024).

(Conneau et al., 2020a). Wu and Dredze (2020) demonstrate that multilingual training leads to lower performance on low-resource languages than higher-resourced ones. Chapter 4 finds that multilingual models forget information previously learned during training, likely due to this development of this phenomenon; Wang et al. (2020) similarly suggest that this effect occurs due to training dynamics. More recently, Chang et al. (2023) presented a controlled study of the factors causing this *curse* that corroborates limited model capacity as the underlying cause. A primary motivation of the work presented in this dissertation is to better understand and limit the effect of this *curse* while maintaining the other benefits of multilingual modeling.

Cross-lingual Transfer in Multilingual Models An unexpected trait of multilingual language models is that they are able to perform cross-lingual transfer, or apply knowledge that they learn in one language to another (Wu and Dredze, 2019). A common application of this skill has been to finetune the model on a task in one language and apply it to a held-out unseen language (e.g., Kondratyuk and Straka, 2019). However, this skill is not universal, as some target languages perform very poorly in this setting (e.g., Lauscher et al., 2020); furthermore, the choice of language pairs between which to transfer information matters (Malkin et al., 2022). This work investigates how cross-lingual transfer develops during the pretraining process (Chapter 4).

Therefore, our experiments contribute to an extensive line of work investigating *how* multilingual LMs perform cross-lingual transfer. Chi et al. (2020a) show that subspaces of mBERT representations that capture syntax are approximately shared across languages, suggesting that portions of the model are cross-lingually aligned. A similar direction of interest is whether multilingual models learn language-agnostic representations. Singh et al. (2019) find that mBERT representations can be partitioned by language, indicating that the representations retain language-specific information. Similarly, other work has shown that mBERT representations can be split into language-specific and language-neutral components (Libovický et al., 2019; Gonen et al., 2020; Muller et al., 2021). However, these works consider the final model checkpoint, while we focus on the development of this cross-lingual phenomenon.

Other work investigated specific factors affecting cross-lingual transfer. These include the effect of sharing subword tokens on cross-lingual transfer (Conneau et al., 2020b; K et al., 2020; Deshpande et al., 2021) and which languages act as good source languages for cross-lingual transfer (Turc et al., 2021). Lauscher et al. (2020), K et al. (2020) and Hu et al. (2020) find that multilingual pretrained models perform worse when transferring to distant languages and low-resource languages; similarly, Martínez-García et al. (2021) show that the morphological typologies of the transfer languages have a strong effect on cross-lingual performance.

Adapting Multilingual Models to New Languages Another common application of multilingual language modeling focuses on how to best adapt existing models to new languages. Initially, these methods continued pretraining these models with the new languages incorporated into the training regime, such as language-adaptive pretraining (LAPT; Chau et al., 2020). Other work proposed the use of adapters to update the model to new languages (Pfeiffer et al., 2020); notably, Faisal and Anastasopoulos (2022) used linguistic information to group languages into adapters — similar to the typological clustering for X-ELMs proposed in Chapter 5. However, follow-up work comparing these prior adaptation approaches found that continued pretraining outperformed adapter methods for new language adaptation (Ebrahimi and Kann, 2021).

Another line of work has considered training or adapting language models with *targeted* or *linguistically informed* multilinguality (Ogueji et al., 2021; Ogunremi et al., 2023; Snæbjarnarson et al., 2023, inter alia). Scao et al. (2022) takes a similar approach by focusing on specific low-resource African languages and other carefully chosen languages. This approach mirrors the inspiration for X-ELM (Chapter 5), where each expert is specialized to a related set of languages (though the full X-ELM still captures a wide variety of languages).

Analysis of Multilingual Knowledge in Language Models There have been several different approaches to quantifying the linguistic information that is learned by multilingual models. One direction has performed layer-wise analyses to quantify what information is stored at different layers in the model (de Vries et al., 2020; Taktasheva et al., 2021; Papadimitriou et al., 2021). Others have

examined the extent to which the different training languages are captured by the model, finding that some languages suffer in the multilingual setting despite the overall good performance exhibited by the models (Conneau et al., 2020a; Wang et al., 2020).

Data analysis, or characterization of the pretraining data, is another approach this work applies to understanding how language models acquire multilingual information; specifically, we look at how multilingual contamination of English corpora teaches "English" language models multilingual skills (Chapter 3). Notably, while most datasets are released with at least some corresponding characterization (more thorough examples include Gao et al. (2020) and Laurençon et al. (2022)), most in-depth analyses are posthoc. These latter works generally assess aspects other than language or multilinguality, such as data source and the effects of quality filtering (Gururangan et al., 2022a), domain and quality (Dodge et al., 2021; Longpre et al., 2023), or task data contamination (Dodge et al., 2021; Blevins et al., 2023). An exception is Kreutzer et al. (2022), which analyzes the quality and accuracy of language categorization of multilingual text corpora.

1.1.2 Other Related Methods and Approaches

Probing NLP Models for Linguistic Knowledge Probing is an analysis technique that tests a model for latent features of interest (Belinkov et al., 2020). There are generally two classes of probing techniques: *structural probing*, or small, linear models that recover implicit attributes of text (e.g., part-of-speech) from the target model's internal vector representations, and *behavioral probing*, which uses carefully designed inputs to test the target model for specific knowledge or skills. In Chapter 2, we present earlier work that proposes structural probes to test NLP models trained on different tasks for syntactic knowledge; this chapter builds on related approaches presented in Shi et al. (2016); Belinkov et al. (2017a,b). Since then, structural probing has become very popular for studying pretrained language models (e.g., Liu et al., 2019a; Tenney et al., 2019), including multilingual ones (e.g., de Vries et al., 2020); we use this approach as well to analyze the models considered in Chapters 3 and 4.

Language Model Training Dynamics Another approach to understanding pretrained LMs has focused on probing multiple checkpoints taken from different points in the pretraining process to quantify when the model learns information. These works have examined the acquisition of syntax (Pérez-Mayos et al., 2021) as well as higher-level semantics and world knowledge over time (Liu et al., 2021) from the RoBERTa pretraining process. Similarly, Chiang et al. (2020) perform a similar temporal analysis for AlBERT, and Choshen et al. (2022) find that the order of linguistic acquisition during language model training is consistent across model sizes, random seeds, and LM objectives.

Most work on probing pretrained models across the training process has focused on monolingual English models. There are some limited exceptions: Dufter and Schütze (2020) present results for multilingual learning in a synthetic bilingual setting, and Wu and Dredze (2020) examine performance across pretraining epochs for a small set of languages. In contrast, this dissertation reports a comprehensive analysis of monolingual and cross-lingual knowledge acquisition on a large-scale multilingual model (Chapter 4).

Sparse Language Modeling The language modeling approach proposed in Chapter 5, X-ELM, builds on existing sparse modeling literature. In general, sparsely activated language models (Evci et al., 2020; Mostafa and Wang, 2019; Dettmers and Zettlemoyer, 2019) route inputs through a subset of the total model parameters. X-ELM builds most directly on the Branch-Train-Merge (BTM) (Li et al., 2022; Gururangan et al., 2023) algorithm, which results in full-model experts trained to specialize on domains of data defined by metadata or a learned clustering. This design expands both on the independent feed-forward network experts found in early Mixture-of-Experts (MoE) models (Jacobs et al., 1991) and on DEMix layers (Gururangan et al., 2022b), which routes sequences to per-layer feed-forward experts based on metadata. Further details about the BTM algorithm are given in §5.2.

Other MoE models have recently been applied to multilingual settings. Pfeiffer et al. (2022) develop a multilingual MoE model with language-specific routing, and Kudugunta et al. (2021)

develop a machine translation model with routing determined by the source-target language pair or the target language. More similarly to BTM, Jang et al. (2023) trains experts specialized to different tasks, including five machine-translation language pairs, which can be merged with other task experts.

1.2 Approach and Roadmap

The two overarching motivations of this dissertation are to (1) establish a better understanding of the current state and limitations of multilingual language models and (2) develop new multilingual learning approaches that work well for every language. To address both of these motivations, this work takes a scientific approach to language technology development. We first perform analyses across many aspects of these technologies — their data, training, and the resulting models — to better understand the current state of multilingual LMs; we then design new modeling approaches that solve issues found during the analysis work.

The remainder of this section provides a roadmap detailing the specific approaches, findings, and contributions of this dissertation:

Probing Models for Syntactic Knowledge Chapter 2 presents the *structural probing* technique used in the following chapters to test multilingual models for linguistic knowledge. This chapter also covers experiments testing earlier RNN-based NLP models (developed before pretraining) for syntactic features within their parameters. We find that RNNs trained on many different NLP tasks — including language modeling — implicitly learn syntax features from these training signals. We also note that one considered training signal is machine translation, thus providing early evidence of the cross-lingual learning of implicit linguistic features. The material in this chapter is adapted from:

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Language Contamination in Pretrained Models Chapter 3 studies how the data used to train language models affects their multilingual knowledge. Specifically, we diagnose how models trained to perform language modeling in a single language (i.e., English) learn to generalize to (and produce linguistic knowledge about) other languages. Our audit of the many popular English text corpora finds language contamination in all cases, despite data filtering to remove non-English text. We then probe models trained on these data for information in other languages, confirming that leaking more data in a target language (unsurprisingly) correlates with model knowledge of that language. These findings indicate (1) that LMs can learn significant information about a language from unexpectedly small subsets of their training data and (2) that automatic data-cleaning processes cannot adequately remove signals about undesired languages. These two factors mean that language models trained at scale will be multilingual. The material in this chapter is adapted from:

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Pretraining Dynamics of Multilingual Language Models Chapter 4 quantifies when multilingual language models acquire knowledge during pretraining. We study this by applying *probing* techniques to model checkpoints from different steps in the pretraining process; this allows us to build a trajectory of mono- and cross-lingual model knowledge over time. We observe that while there are some overall trends in multilingual learning,[4] knowledge acquisition speed and model performance vary greatly across languages and language pairs; furthermore, these differences often are unpredictable from language-specific factors. A specific, unexpected behavior observed across languages is model forgetting, where model knowledge becomes *worse* during pretraining for certain languages and skills; we hypothesize that this behavior is due to languages competing for model capacity during model optimization.[5] The material in this chapter is adapted from:

[4]Such as that linguistic knowledge is acquired in a hierarchical order, and that monolingual information is obtained before cross-lingual skills.

[5]And leading to the *curse of multilinguality* as observed in the final checkpoints of multilingual LMs.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono- and cross-lingual pretraining dynamics of multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Cross-lingual Expert Language Models The prior two chapters demonstrated that though LMs can learn significant amounts of information about many languages from limited data, current approaches to pretraining do not allow these models to fully leverage that data in massively multilingual training. Chapter 5 proposes X-ELMS, or Cross-lingual Expert Language Models, as a new method for multilingual modeling that better represents different languages by explicitly allocating model capacity to them. X-ELMs consist of a set of language models, where each LM, or expert, is specialized to a subset of a multilingual space using the Branch-Train-Merge (BTM) paradigm (Li et al., 2022); they can then be used either individually or as an ensemble to perform the full scope of multilingual modeling. This approach improves LM performance for all of the considered languages and gives training efficiency and model adaptation benefits over current multilingual modeling approaches. These improvements mean that X-ELMs are a step towards democratizing multilingual NLP: they both allow for better performance on lower-resourced languages and are easier to train and use in low-compute settings. The material in this chapter is adapted from:

Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A. Smith, and Luke Zettlemoyer. 2024. Breaking the curse of multilinguality with cross-lingual expert language models. *arXiv preprint arXiv:2401.10440*.

Conclusion The final portion of this dissertation summarizes the overall findings (Chapter 6). We conclude by discussing next steps towards further understanding and improving multilinguality in NLP, focusing on better data collection practices and modeling for under-resourced languages and communities.

Chapter 2

Probing Models for Syntactic Knowledge

We present a set of experiments to demonstrate that deep recurrent neural networks (RNNs) learn internal representations that capture soft hierarchical notions of syntax from highly varied supervision. We consider four syntax tasks at different depths of the parse tree; for each word, we predict its part of speech as well as the first (parent), second (grandparent) and third level (great-grandparent) constituent labels that appear above it. These predictions are made from representations produced at different depths in networks that are pretrained with one of four objectives: dependency parsing, semantic role labeling, machine translation, or language modeling. In every case, we find a correspondence between network depth and syntactic depth, suggesting that a soft syntactic hierarchy emerges. This effect is robust across all conditions, indicating that the models encode significant amounts of syntax even in the absence of explicit syntactic supervision.[1]

2.1 Introduction

Deep recurrent neural networks (RNNs) have effectively replaced explicit syntactic features (e.g. parts of speech, dependencies) in state-of-the-art NLP models (He et al., 2017; Lee et al., 2017; Klein et al., 2017). However, previous work has shown that syntactic information (in the form of either input features or supervision) is useful for a wide variety of NLP tasks (Punyakanok et al.,

[1]This chapter includes materials originally published in Blevins et al. (2018).

2005; Chiang et al., 2009), even in the neural setting (Aharoni and Goldberg, 2017; Chen et al., 2017). In this chapter, we show that the internal representations of RNNs trained on a variety of NLP tasks encode these syntactic features without explicit supervision.

We consider a set of feature prediction tasks drawn from different depths of syntactic parse trees; given a word-level representation, we attempt to predict the POS tag and the parent, grandparent, and great-grandparent constituent labels of that word. We evaluate how well a simple feed-forward classifier can detect these syntax features from the word representations produced by the RNN layers from deep NLP models trained on the tasks of dependency parsing, semantic role labeling, machine translation, and language modeling. We also evaluate whether a similar classifier can predict if a dependency arc exists between two words in a sentence, given their representations.

We find that, across all four types of supervision, the representations learned by these models encode syntax beyond the explicit information they encounter during training; this is seen in both the word-level tasks and the dependency arc prediction task. Furthermore, we also observe that features associated with different levels of syntax tree correlate with word representations produced by RNNs at different depths. Largely speaking, we see that deeper layers in each model capture notions of syntax that are higher-level and more abstract, in the sense that higher-level constituents cover a larger span of the underlying sentence.

These findings suggest that models trained on NLP tasks are able to induce syntax even when direct syntactic supervision is unavailable. Furthermore, the models are able to differentiate this induced syntax into a soft hierarchy across different layers of the model, perhaps shedding some light on why *deep* RNNs are so useful for NLP.

2.2 Methodology

Given a model that uses multi-layered RNNs, we collect the vector representation $\mathbf{x}_i^l$ of each word i at each hidden layer l. To determine what syntactic information is stored in each word vector, we try to predict a series of constituency-based properties from the vector alone. Specifically, we

predict the word's part of speech (POS), as well as the first (parent), second (grand-parent), and third level (great-grandparent) constituent labels of the given word. Figure 2.1 shows how these labels correspond to an example constituency tree.

Our methodology follows Shi et al. (2016), who run syntactic feature prediction experiments over a number of different shallow machine translation models, and Belinkov et al. (2017a,b), who use a similar process to study the morphological, part-of-speech, and semantic features learned by deeper machine translation encoders. We extend upon prior work by considering training signals for models other than machine translation, and by applying more stratified word-level syntactic tasks.

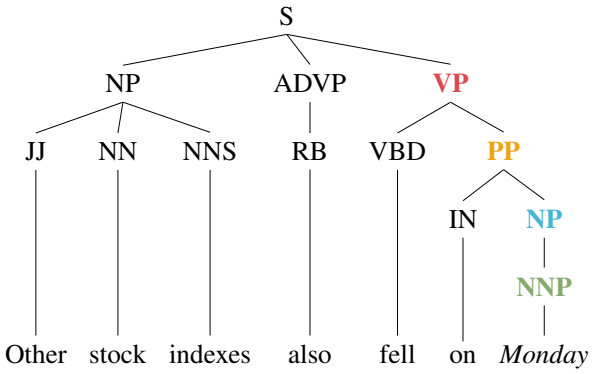


Figure 2.1: Constituency tree with labels for the word "Monday" for the POS (green), parent constituent (blue), grandparent constituent (orange), and great-grandparent constituent (red) tasks.

Training Signal	**Dataset**	**RNN Layers**	**Input Dims**	**Hidden Dims**
Dependency Parsing	English Universal Dependencies	4 parallel bidirectional LSTMs	200	400
Semantic Role Labeling	CoNLL-2012	8 alternating-direction LSTMs with highways	100	300
Machine Translation	WMT-2014 English-German	4 parallel bidirectional LSTMs	500	500
Language Modeling	CoNLL-2012	2 sets of 4 unidirectional LSTMs with highways	1000	1000

Table 2.1: The training data, recurrent architecture, and hyperparameters of each model.

2.2.1 Experiment Setup

We predict each syntactic property with a simple feed-forward network with a single 300-dimensional hidden layer activated by a ReLU:

$$\mathbf{y}_i^l = \text{SoftMax}(W_2\text{ReLU}(W_1\mathbf{x}_i^l)) \tag{2.1}$$

where i is the word index and l is the layer index within a model. To ensure that the classifiers are not trained on the same data as the RNNs, we train the classifier for each layer l separately using the development set of CoNLL-2012 and evaluate on the test set (Pradhan et al., 2013).

In addition, we compare performance with word-level baselines. We report the per-word majority class baseline; at the POS level, for example, "cat" will be classified as a noun and "walks" as a verb. This baseline outperforms the pre-trained GloVe (Pennington et al., 2014) embeddings on every task. We also consider a contextual baseline, in which we concatenate each word's embedding with the average of its context's embeddings; however, this baseline also performed worse than the reported one.

2.2.2 Analyzed Models

We consider four different forms of supervision. Table 2.1 summarizes the differences in data, architecture, and hyperparameters.[2]

Dependency Parsing We train a four-layer version of the Stanford dependency parser (Dozat and Manning, 2017) on the Universal Dependencies English Web Treebank (Silveira et al., 2014). We ran the parser with 4 bidirectional LSTM layers (the default is 3), yielding a UAS of 91.5 and a weighted LAS of 82.18, consistent with the state of the art on CoNLL 2017. Since the parser receives syntactic features as input (POS) and is trained on an explicit syntactic signal, we expect

[2]While we control for some variables, we mainly rely on existing architectures and hyperparameters that were tuned for the original tasks, limiting the cross-model comparability of absolute performance levels on our syntactic evaluations.

its intermediate representations to contain a high amount of syntactic information.

Semantic Role Labeling We use the pre-trained DeepSRL model from (He et al., 2017), which was trained on the training data from the CoNLL-2012 dataset. This model is an alternating bidirectional LSTM, where the model consists of eight total layers that alternate between a forward layer and backward layer. We concatenate the representations from each pair of directional layers in the model for consistency with other models.

Machine Translation We train a machine translation model using OpenNMT (Klein et al., 2017) on the WMT-14 English-German dataset. The encoder (which we examine in our experiments) is a 4-layer bidirectional LSTM; we use the defaults for every other setting. The model achieves a BLEU score of 21.37, which is in the ballpark of other vanilla encoder-decoder attention models on this benchmark (Bahdanau et al., 2015).

Language Modeling We train two separate language models on CoNLL-2012's training set, one going forward and another backward. Each model is a 4-layer LSTM with highway connections, variational dropout, and tied input-output embeddings. After training, we concatenate the forward and backward representations for each layer.[3]

2.3 Constituency Label Prediction

Figure 2.2 summarizes our findings, while Table 2.2 presents the full numerical results. We make several observations:

RNNs can induce syntax. Overall, each model outperforms the baseline and its respective input embeddings on every syntax task, indicating that their internal representations encode some notions of syntax. The only exception to this observation is POS prediction with dependency parsing

[3]The model achieved perplexities of 50.56 (forward) and 51.24 (backward) on CoNLL-2012's test set. Since we are not familiar with other perplexity results on this data, we note that retraining the architecture on Penn TreeBank achieved 64.39 perplexity, which is comparable to other high-performing language models.

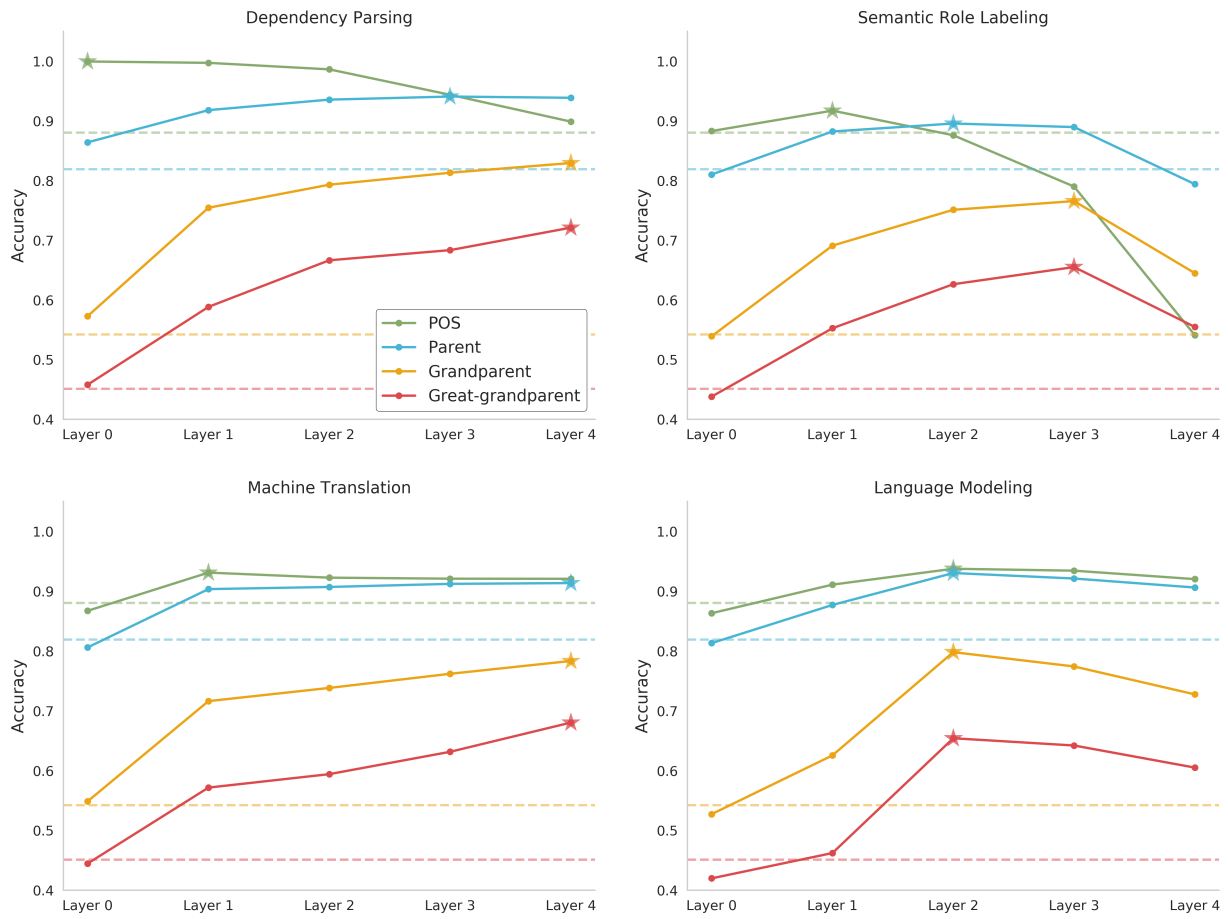


Figure 2.2: Results of syntax experiments. The best-performing layer for each experiment is annotated with a star, and the per-word majority baseline for each task is shown with a dashed line.

representations; in this case the parser is provided gold POS tags as input, and cannot improve upon them. This result confirms the findings of Shi et al. (2016) and Belinkov et al. (2017b), who demonstrate that neural machine translation encoders learn syntax, and shows that RNNs trained on other NLP tasks also induce syntax.

Deeper layers reflect higher-level syntax. In 11 out of 16 cases, performance improves up to a certain layer and then declines, suggesting that the deeper layers encode less syntactic information that earlier ones in these cases. Strikingly, the higher-level a syntactic task is, the deeper in the network the peak performance occurs; for example, in SRL we see that the parent constituent task peaks one layer after POS, and the grand-parent and great-grandparent tasks peak on the layer

			Layer				
Source model	Prediction task	MFT	0	1	2	3	4
DP	POS	0.8801	**0.9990**	0.9964	0.9853	0.9434	0.8962
	Parent	0.8190	0.8681	0.9177	0.9347	**0.9384**	0.9349
	Grandparent	0.5422	0.5721	0.7538	0.7920	0.8094	**0.8253**
	Great-Grandparent	0.4511	0.4648	0.5909	0.6659	0.6826	**0.7183**
SRL	POS	0.8801	0.8732	**0.9063**	0.8691	0.7833	0.5392
	Parent	0.8190	0.7983	0.8727	**0.8892**	0.8835	0.7870
	Grandparent	0.5422	0.5041	0.6812	0.7394	**0.7549**	0.6325
	Great-Grandparent	0.4511	0.4412	0.5415	0.6159	**0.6449**	0.5493
MT	POS	0.8801	0.8618	**0.9274**	0.9198	0.9191	0.9195
	Parent	0.8190	0.8019	0.8975	0.9040	**0.9088**	0.9083
	Grandparent	0.5422	0.5368	0.7072	0.7311	0.7572	**0.7776**
	Great-Grandparent	0.4511	0.4361	0.5631	0.5909	0.6303	**0.6752**
LM	POS	0.8801	0.8608	0.9093	**0.9359**	0.9304	0.9165
	Parent	0.8190	0.8126	0.8724	**0.9232**	0.9137	0.9000
	Grandparent	0.5422	0.5237	0.6251	**0.7862**	0.7606	0.7189
	Great-Grandparent	0.4511	0.4249	0.4702	**0.6423**	0.6302	0.5971

Table 2.2: Table of accuracy results for the syntax feature prediction experiments with best performing layer in each source model/ prediction task pair in bold. "DP" refers to the dependency parsing model.

after that. One possible explanation is that each layer leverages the shallower syntactic information learned in the previous layer in order to construct a more abstract syntactic representation. In SRL and language modeling, it seems as though the syntactic information is then replaced by task-specific information (semantic roles, word probabilities), perhaps making it redundant.

This observation may also explain a modeling decision in ELMo (Peters et al., 2018), where injecting the contextualized word representations from a pre-trained language model was shown to boost performance on a wide variety of NLP tasks. ELMo represents each word using a task-specific weighted sum of the language model's hidden layers, i.e. rather than use only the top layer, it selects which of the language model's internal layers contain the most relevant information for the task at hand. Our results confirm that, in general, different types of information manifest at different layers, suggesting that post-hoc layer selection can be beneficial.

Language models learn some syntax. We compare the performance of language model representations to those learned with dependency parsing supervision, in order to gauge the amount of syntax induced. While this comparison is not ideal (the models were trained with slightly different architectures and hyperparameters), it does provide evidence that the language model's representations encode some amount of syntax implicitly. Specifically, we observe in Figure 2.3 that the language model and dependency parser perform nearly identically on the three constituent prediction tasks in the second layer of their respective networks. In deeper layers the parser continues to improve, while the language model peaks at layer 2 and drops off afterwards.

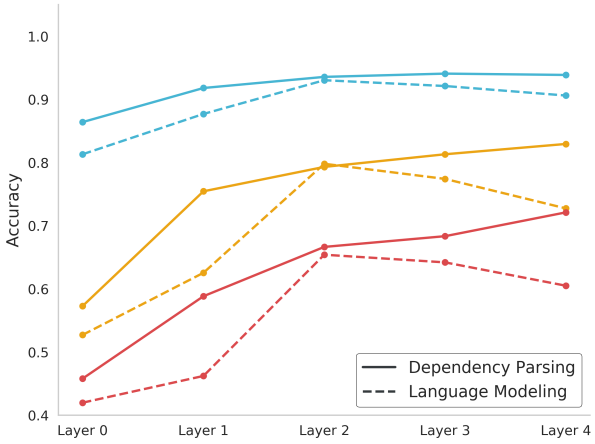


Figure 2.3: Comparison between the LM and dependency parser on the parent (blue), grandparent (yellow), and great-grandparent (red) constituent prediction tasks.

These results may be surprising given the findings of Linzen et al. (2016), which found that RNNs trained on language modeling perform below baseline levels on the task of subject-verb agreement. However, the more recent investigation by Gulordava et al. (2018) are in line with our results. They find that language models trained on a number of different languages assign higher probabilities to valid long-distance dependencies than to incorrect ones. Therefore, LMs seem able to induce syntactic information despite being provided with no linguistic annotation.

2.4 Dependency Arc Prediction

We run an additional experiment that seeks to clarify if the representations learned by deep NLP models capture information about syntactic structure. Using the internal representations from a deep RNN, we train a classifier to predict whether two words share an dependency arc (have a

parent-child relationship) in the in the dependency parse tree over a sentence. We find that, similarly to the previous set of tasks, deep RNNs trained on various linguistic signals encode notions of the syntactic relationships between words in a sentence.

Setup We use the same pretrained deep RNNs and feed-forward prediction network paradigm. However, we change the input from the previous experiments, as this task is not at the word-level, but rather concerns the relationship between two words; therefore, given a word pair w_c, w_p for which we have a dependency arc label, we input $[w_c; w_p; w_c \circ w_p]$ into the classifier.

We use the Universal Dependencies dataset for this task, such that we train each classifier on the development set of this dataset and evaluate on the test set. We set up the task by generating two pairs of examples for each word in the UD dataset: a positive pair that consists of the word and its parent in the dependency tree, and a negative pair that matches the word with another randomly chosen word from the sentence.

Source Model	GloVe	L0	L1	L2	L3	L4
DP	0.50	0.68	0.77	0.81	0.88	**0.95**
SRL	0.50	0.58	0.69	0.76	**0.79**	0.74
MT	0.50	0.61	**0.73**	0.63	0.63	0.63
LM	0.50	0.62	0.74	0.78	**0.80**	0.73

Table 2.3: Results of the dependency arc prediction task. L0–L4 denote the different layers of the model. DP refers to the RNN trained with dependency parsing supervision.

Results The results for this prediction task are given in Table 2.3. We see the best performance from the dependency parser, finding that the performance for the dependency parser's representations continue to improve in the deepest layers, with a maximum performance of approximately 95% on the last layer. This result is unsurprising, as this closely related to the task on which the model was explicitly trained. In the three other models, we find peaks that occur 12 to 20 accuracy points above the input layer's performance. These results support the findings from the constituency label prediction task and show that these findings hold up across syntactic formalisms.

Similarly to the word-level tasks, we see the best performance from deeper layers in the models, with both SRL and LM performance peaking on the third layer. For the LM, we find that the best performing layer outperforms the initial layer by 18%. This is consistent with our finding in the previous set of experiments, that RNNs encode significant amounts of syntax information even when trained on linguistic tasks without any explicit annotations.

2.5 Conclusions

In this chapter, we run a series of prediction tasks on the internal representations of deep NLP models, and find these RNNs are able to induce syntax without explicit linguistic supervision. We also observe that the representations taken from deeper layers of the RNNs perform better on higher-level syntax tasks than those from shallower layers, suggesting that these recurrent models induce a soft hierarchy over the encoded syntax. These results provide some insight as to why deep RNNs are able to model NLP tasks without annotated linguistic features. Further characterizing the exact aspects of syntax which these models can capture (and perhaps more importantly, those they cannot) is an interesting area for future work.

Chapter 3

Language Contamination in Pretrained Models

English pretrained language models, which make up the backbone of many modern NLP systems, require huge amounts of unlabeled training data. These models are generally presented as being trained only on English text but have been found to transfer surprisingly well to other languages. We investigate this phenomenon and find that common English pretraining corpora actually contain significant amounts of text in other languages: even when less than 1% of data is not English (well within the error rate of strong language classifiers), this leads to hundreds of millions of foreign language tokens in large-scale datasets. We then demonstrate that even these small percentages of non-English data facilitate cross-lingual transfer for models trained on them, with target language performance strongly correlated to the amount of in-language data seen during pretraining. In light of these findings, we argue that no model is truly monolingual when pretrained at scale, which should be considered when evaluating cross-lingual transfer. [1]

[1]This chapter includes materials originally published in Blevins and Zettlemoyer (2022). Further details and experiments are given in Appendix A.

3.1 Introduction

Pretrained language models have become an integral part of NLP systems. They come in two flavors: *monolingual*, where the model is trained on text from a single language, and *multilingual*, where the model is jointly trained on data from many different languages. Monolingual pretrained models are generally applied to tasks in the same language, whereas multilingual ones are used for cross-lingual learning.

Recent work has claimed that monolingual pretrained models are also surprisingly good at transferring between languages, despite ostensibly having never seen the target language before (Gogoulou et al., 2021; Li et al., 2021, inter alia). However, because of the large scale of pretraining data and because many pretraining corpora are not publicly available, it is currently unknown how much foreign language data exists in monolingual pretraining corpora. In this Section, we show that (1) these data are almost certainly contaminated with very small percentages of text from other languages and that (2) cross-lingual transfer is possible from such data leakage in the pretraining corpus.

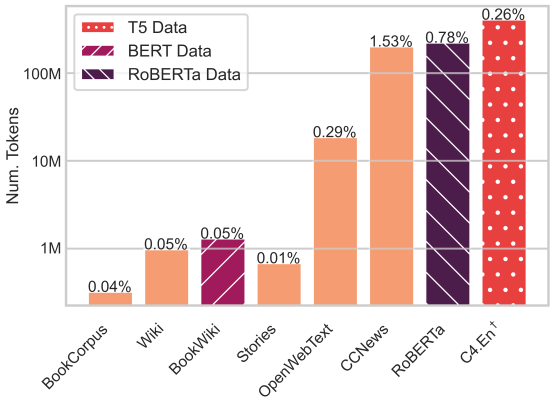


Figure 3.1: Estimated non-English data in English pretraining corpora (token count and total percentage); even small percentages lead to many tokens. C4.En (†) is estimated from the first 50M examples in the corpus.

More specifically, we quantify how *multilingual* English pretrained models are in two steps. First, we analyze common English pretraining corpora with a large-scale automatic evaluation to estimate their language composition, as well as a smaller-scale manual analysis. Second, we perform experiments across fifty languages on masked language modeling and part-of-speech (POS) tagging to measure how well the models trained on these pretraining corpora perform outside of English.

Our analysis finds that these corpora include very small percentages that amount to overall significant amounts of text in other languages (Figure 3.1), particularly those derived from web-

crawled data. Furthermore, the models trained on this data perform surprisingly well on other languages; this transfer is strongly correlated with the amount of target language data seen during pretraining. Notably, we find that the English T5 outperforms mBERT on POS tagging in multiple languages with no finetuning.

Overall, these results indicate that the considered models are actually multilingual and that their ability to transfer across languages is not zero-shot, despite what has been recently claimed. Given the effort required to fully remove all non-English data, we question whether it is practically possible to train truly monolingual models at scale.

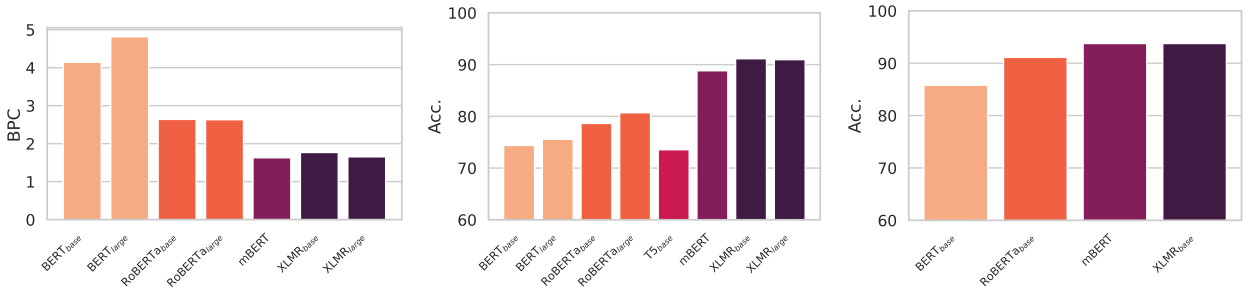
3.2 Pretraining Data Composition

We first measure the amount of non-English text in commonly used English pretraining corpora with two analyses: automatic language identification to estimate the amount of foreign language data in these corpora and a manual qualitative analysis of the text classified as a language other than English.

We consider the following pretraining datasets: ENGLISH WIKIPEDIA (11.8GB); BOOKCORPUS (Zhu et al. 2015, 4.2GB); STORIES (Trinh and Le 2018, 31GB); OPENWEBTEXT (Gokaslan and Cohen 2019, 38GB), which is an open-source version of WEBTEXT (Radford et al., 2019); CC-NEWS (Liu et al. 2019b, 76 GB); and C4.EN (Raffel et al. 2020, 305GB), as provided by Dodge et al. (2021). We use the versions of WIKIPEDIA, BOOKCORPUS, and CC-NEWS used to pretrain RoBERTa.

3.2.1 Automatic Evaluation of Language Composition

We use the FastText language identification model (Joulin et al., 2017) to label every line in each corpus and consider lines to be non-English if they score above a set confidence threshold (0.6). Due to the large size of C4, we subsample the first 50M examples (or 14%); we classify the entirety of all other datasets. Since language detection is imperfect, particularly for low-resource languages



(a) MLM **(b)** POS (probing) **(c)** POS (finetuned)

Figure 3.2: Average performance by each model across all languages for the task. Lower is better for BPC.

(Kreutzer et al., 2022), we present the results of this analysis as an estimate of the non-English data in each dataset and perform a qualitative analysis of potential errors in the following section.

A summary of the language identification experiments is presented in Figure 3.1. We see that every corpus contains notable quantities of non-English data, with our estimates ranging between 300k to 406M tokens. An obvious factor that affects the amount of non-English data in each corpus is the overall size of the dataset; however, even when controlling for size by looking at the percentage of non-English data, we still see that the smaller corpora (WIKIPEDIA, BOOKCORPUS, and STORIES) have relatively less leakage in their data.

Indeed, a major factor of language leakage is the method in which the data was collected: the datasets derived from web crawls contain higher percentages of text in other languages (OPENWEBTEXT andCCNEWS). This is true even for C4, where the dataset was filtered with a classifier to only include English text (Raffel et al., 2020). Since automatic methods for language identification are imperfect, the datasets with more manual filtering (such as WIKIPEDIA, which has human editors curating its content) are less prone to language contamination than those relying on classifiers. Due to these challenges, it is likely impossible to fully filter text corpora by language at scale.

We also see that non-English text makes up small percentages of the overall data, though this still leads to millions of tokens in large datasets. The largest individual languages after English only make up 0.01%, 0.15%, and 0.05% of the BERT, RoBERTa, and T5 training data, respectively. Multilingual pretraining work has shown that models generalize to new languages from varying

amounts of data (Delvin, 2019; Conneau and Lample, 2019; Conneau et al., 2020a); however, these approaches intentionally select data across languages, and most upsample low-resource languages during training. Without these considerations, it is an open question how well the models trained on these relatively small amounts of data in other languages generalize.

Type	Num. of Lines in...					
	Book	**Wiki**	**Stories**	**OpenWeb**	**CCNews**	**C4**
NE	156	129	99	175	193	169
	Ex: Moraliska argument utgår ifrån våra moraliska intuitioner att rätt och fel inte endast är förankrade i människors vilja. (OPENWEBTEXT)					
BiL	13	11	15	4	1	22
	Ex: The German blazon reads: "Von Silber über Schwarz geteilt..." (WIKI)					
Trans.	2	7	4	2	0	4
	Ex: Εκείνη δεν μπορούσε να πληρώσει [She couldn't pay.] (BOOKCORPUS)					
Ent.	1	28	5	1	0	1
	Ex: 2012 Playhouse Presents ウィルシリーズ1、エピソード1: "The Minor Character" (C4)					
En	26	22	55	12	6	3
	Ex: "Dere's buzzards circlin' ova dem trees." (BOOKCORPUS)					
XX	2	3	22	6	0	1
	Ex: M D I X O X I O O O = A (WIKI)					

Table 3.1: Results of the qualitative analysis of the non-English lines in various pretraining corpora. Type abbreviations are defined in §3.2.2.

3.2.2 Qualitative Analysis of Language Contamination

We also perform a closer analysis on a random subset (200 per corpus) of lines predicted by the language classifier to be not English text (Table 3.1). Each example is manually coded into one of six categories. The first set covers various kinds of foreign language data: **NE** (*non-English*), where the line contains only text in a language other than English; **BiL**, or *bilingual*, where the line contains both English text and text in a different language; **Trans.**, in which the English and other texts are *translations* of each other; and **Ent.**, where the line is primarily English but contains *entities* in a different language. The last two codes pertain to errors made by the language classifier:

En., where the line only contains *English* text—as expected in filtered, English corpora—and **XX**, which refers to lines that contain *no natural language*.

The majority of lines across datasets consist entirely of text in a language other than English. The next most common type of contaminated data is **BiL**; this contains many subtypes of data, such as codeswitching and foreign language dialogue within English text. These datasets also include parallel data at both the sentence- and word-level.[2] We note that all observed translations are between English and another language. Finally, some of the examples classified as non-English are actually English texts containing phrases in other languages.

Our analysis also shows that the language classifier performs worse on the non-web crawled data. For example, it misclassified a quarter of the sampled lines from STORIES as non-English when they in fact only contain English text; many of these lines stem from snippets of dialogue in the dataset. We generally observe that lines coded as **En** tend to be shorter than the correctly labeled lines and often contain non-standard English. The language classifier also struggles to handle noisy lines, for which it has no appropriate language label.

3.3 Cross-lingual Transfer of English Pretrained Models

We now ask: how well do models pretrained on these putatively English corpora perform on tasks in other languages? While the English data is more multilingual than previously thought, there are many differences between monolingual and multilingual pretraining; these data from other languages are often tokenized into more subword units[3] and are much less frequently observed during monolingual training. Furthermore, Magar and Schwartz (2022) find that models do not always learn to exploit contaminated components of their data, indicating that presence in the pretraining data does not guarantee downstream model performance.

[2]e.g., "大学【だい・がく】– college", OPENWEBTEXT

[3]For example, the Basque UD treebank requires on average 1.78, 2.59, and 2.66 tokens per word to be encoded by XLMR, RoBERTa, and BERT, respectively.

3.3.1 Experimental Setup

We evaluate popular English pretrained models on tasks in more than 50 languages: (masked) language modeling, POS probing, and finetuned POS tagging. We compare the performance of monolingual BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), and T5 (Raffel et al., 2020) against multilingual mBERT (Delvin, 2019) and XLM-R (Conneau et al., 2020a). We report average performance across five runs with different random seeds for the POS evaluations.

For the language modeling experiments, we perform whole word masking on 15% of the words in the Wiki40B test set to calculate BPC. This experiment was zero-shot and required no further training of the models.

For the POS probing experiments, we train a linear classifier to predict POS from the final layer of each considered encoder; each probe therefore consists of a limited number of parameters $m * l$ where m is the output dimension of the encoder being probed (768 for base models and 1024 for large models) and l is the size of the label set (17 for POS tagging). For words that are tokenized into multiple subword units, we take the average representation of all tokens as the input to the classifier. When finetuning the model, we take the same setup as probing but unfreeze the encoder weights to allow them to update during training. The POS models are trained and evaluated on Universal Dependencies (UD) treebanks for each language (Nivre et al., 2020).

We use a batch size of 256 for the frozen experiments and batch sizes of 16 for the finetuned models; we used a learning rate of 0.001 for the probing task and 5e-6 for finetuning. Due to the large number of experiments, we did not tune these parameters. For both POS tagging experiments, we use an Adam optimizer (Kingma and Ba, 2015), and train each probe for 50 passes over the data (with early stopping on the validation set and a patience of 5). The pretrained models for all experiments are downloaded from Huggingface (Wolf et al., 2019).

Each of our models was trained on a single Nvidia V100 GPU: 16GB for the frozen models and 32GB for the finetuned ones. The frozen probes each took between <1 and 8 minutes to train, and the finetuned probes were trained for between 5 minutes and 7.5 hours (depending on the dataset size, which varies by language, and early stopping epoch).

3.3.2 Multilingual MLM Evaluation

We first measure the perplexity of English pretrained MLMs in other languages. We use Wiki-40B, a multilingual language modeling dataset that covers 41 languages (Guo et al., 2020). Following the Wiki-40B paper, we report bits per character (BPC) to allow comparison between models with different tokenizations of the text.

We find that both BERT models perform notably worse on modeling other languages; however, RoBERTa, reduces the gap with the multilingual models from 2.51 BPC to 0.87 BPC (Figure 3.2a). This finding is consistent with Tran (2020), who also found RoBERTa transfers well cross-lingually.

3.3.3 POS Performance Across Languages

Next, we evaluate how well monolingual English models perform on multilingual downstream tasks, using part-of-speech (POS) tagging as a case study.

Probing We first consider the performance of the encoders when probed for POS knowledge using the linear probing method discussed in Chapter 2 (Figure 3.2b).[4] Unsurprisingly, on average all of the English models underperform the multilingual models. Similar to MLM, we find that RoBERTa performs better than BERT when probed for POS features on other languages; surprisingly, it also strongly outperforms T5, despite C4 containing more absolute non-English data than the RoBERTa corpus.

This difference is likely due to two factors. First, in terms of relative percentages, RoBERTa is exposed to more non-English text than T5 (0.78% compared to only 0.22%). Secondly, RoBERTa's subword vocabulary is robust to unexpected inputs and does not substitute an UNK token any input tokens; in contrast, T5 and BERT have high rates of UNK tokens for some non-Latin languages.[5] However, for many high-resource languages the English models perform competitively, with T5 outperforming mBERT on German and Portuguese, among others.

[4] For T5, this means that we evaluate the output of the encoder and discard the decoder.

[5] *UNK tokens* refer to placeholder tokens used when the model receives an input not covered by its vocabulary.

Task	Model	Corr. (ρ) with...	
		lang. data ↑	en sim. ↓
MLM (BPC) ↓	BERT_{base}	-0.258	0.097
	BERT_{lg}	-0.258	0.118
	RoBERTa_{base}	-0.667**	0.326*
	RoBERTa_{lg}	-0.685**	0.345*
Frozen POS (Acc.) ↑	BERT_{base}	0.335*	-0.332*
	BERT_{lg}	0.314*	-0.375*
	RoBERTa_{base}	0.594**	-0.260
	RoBERTa_{lg}	0.674**	-0.304*
	T5_{base}	0.131	-0.271
Finetuned POS (Acc.) ↑	BERT_{base}	0.373*	-0.340*
	RoBERTa_{base}	0.507**	-0.292*

Table 3.2: Spearman correlations between task performance and (a) in-language data amounts in pretraining corpora (*lang. data*) and (b) language similarity with English (*en sim.*). $^{*}p < 0.05$ and $^{**}p < 0.001$.

Fine-tuning To test if the effects of foreign language data carry through after finetuning, we also finetune a subset of the models (BERT_{base}, RoBERTa_{base}, mBERT, XLMR_{base}) for multilingual POS tagging (Figure 3.2c). After finetuning, the gap between the mono- and multilingual models is much smaller: RoBERTa only averages 2.65 points worse than XLM-R, compared to 12.5 points when probing.

3.3.4 Potential Reasons for Cross-lingual Generalization

We then investigate the correlation between potential transfer causes and model performance (Table 3.2). Specifically, we consider the quantity of target language data found in the model's pretraining corpus and the language similarity to English as potential causes of cross-lingual transfer.

We find that across tasks, RoBERTa task performance is most strongly correlated with the amount of target language data seen during pretraining. BERT and T5 task performance are less correlated with observed pretrained data, which is likely due to tokenization artifacts (§3.4). Indeed, when we control for languages not written with Latin script on T5, the correlation between performance and the amount of target pretraining data increases to $\rho = 0.313$.

We also consider the effect of language similarity on task performance, which is often hypothe-

sized to facilitate cross-lingual transfer. We use the syntactic distance of languages calculated by Malaviya et al. (2017); more similar languages score lower. However, we generally find that this is less correlated with performance than the quantity of target text, particularly for RoBERTa.

3.4 The Effect of Tokenization

A factor that varies across the considered models is how they tokenize the input text for different languages. Appendix Table A.2 gives the number of subword tokens per (white-space separated) word in the validation split of Wiki40b (Guo et al., 2020), as well as the percentage of tokens that are unked out by the tokenizer. We see that in general, all of the models (including explicitly multilingual ones) require more subword tokens per word for languages other than English.[6] We can also see that T5 is more efficient at encoding French, German, and Romanian than the other monolingual models (without a high UNK rate), likely because the T5 tokenizer was explicitly trained on English data mixed with those languages (Raffel et al., 2020).

We also examine how many tokens are unked out by each tokenizer across languages. We see that BERT and T5 in particular have a high UNK rate ($> 10\%$) for many languages not written in Latin script. This is in part due to the different tokenization schemes used by the models: RoBERTa uses a byte-level BPE encoding (Radford et al., 2019), which produces no UNK tokens for Unicode text, whereas the tokenization methods used by BERT and T5 (SentencePiece, Kudo and Richardson (2018)) will unk out tokens not seen while training the tokenizer. Additionally, there are other potential decisions made during tokenization that could affect these UNK rates, including filtering on non-Latin tokens or learning the subword tokenizer on a subset of the training data.

High UNK rates in the tokenized text for a language affect performance on downstream tasks. With regards to evaluating BPC, high frequencies of UNK tokens in the data likely make the language modeling task artificially easy, leading to lower BPC scores. Because of this, we note the cases where a model UNKs out more than 10% of the considered data in the BPC results given in

[6]We note that the number of subword tokens per "word" in Japanese is much larger than in other languages, as words in Japanese are not whitespace-separated.

Appendix Table A.3 with an asterisk (*). High UNK rates likely also lead to degraded performance on downstream tasks (including the considered POS tagging task in this work).

3.5 Discussion

In this Section, we demonstrate that English pretrained models are unintentionally exposed to a considerable amount of multilingual text data during pretraining, particularly in the case of more recent models that are trained on larger corpora derived from web crawls. We also find that this non-English text acts as a significant source of signal for cross-lingual transfer.

Other recent work has focused on documenting the composition of pretraining corpora (Dodge et al., 2021; Gururangan et al., 2022a). Kreutzer et al. (2022) manually audit a variety of multilingual datasets, finding data quality issues that are worse for low-resource languages and, similarly to our work, that texts for many languages are misclassified. In contrast, our focus is on the presence of foreign language data in primarily English corpora.

Prior work has also shown the ability of monolingual models to transfer to other languages across a wide range of tasks (Gogoulou et al., 2021; Li et al., 2021; Tran, 2020; Artetxe et al., 2020; Chi et al., 2020b), but these works do not consider the effect of foreign language data leakage as a source of signal. Notably, de Souza et al. (2021) mention the presence of foreign language data in their corpora but assume the small amounts observed will not affect model performance. However, our findings demonstrate that the amount of foreign language data directly correlates with cross-lingual transfer.

While we focus on English pretraining, many monolingual models have been developed for other languages on similar datasets to the ones we analyze. Additionally, some related work has investigated transferring these pretrained models onto English tasks (e.g. Gogoulou et al., 2021). Given the prevalence of English text on the internet (Pimienta et al., 2009), it is likely that English (and other foreign language) contamination of these datasets is common.

An obvious follow-up to our findings would be to retrain the models with text that is verified to

only contain English data; this would confirm the effect the leaked language data has on the models. We reiterate that the standard method for filtering these datasets, automatic language classifiers, is imperfect. This, and the infeasibility of manual filtering due to the scale of the data, means that controlling for the language the model is pretrained on is nearly impossible.

However, the presence of foreign language data in pretraining corpora is not inherently problematic. Models trained on these datasets perform exceedingly well on their target languages *and* generalize to other languages much better than expected. Rather, it is important to remember that these models are not performing zero-shot transfer when used in other languages, given the scale and data with which they were pretrained.

Chapter 4

Pretraining Dynamics of Multilingual Language Models

The emergent cross-lingual transfer seen in multilingual pretrained models has sparked significant interest in studying their behavior. However, because these analyses have focused on fully trained multilingual models, little is known about the dynamics of the multilingual pretraining process. We investigate *when* these models acquire their in-language and cross-lingual abilities by probing checkpoints taken from throughout XLM-R pretraining, using a suite of linguistic tasks. Our analysis shows that the model achieves high in-language performance early on, with lower-level linguistic skills acquired before more complex ones. In contrast, the point in pretraining when the model learns to transfer cross-lingually differs across language pairs. Interestingly, we also observe that, across many languages and tasks, the final model layer exhibits significant performance degradation over time, while linguistic knowledge propagates to lower layers of the network. Taken together, these insights highlight the complexity of multilingual pretraining and the resulting varied behavior for different languages over time.[1]

[1]This chapter presents a summary of, and includes materials originally published in, Blevins et al. (2022). Appendix B provides further experimental details and results.

4.1 Introduction

Large-scale language models pretrained jointly on text from many different languages (Delvin, 2019; Conneau and Lample, 2019; Lin et al., 2022) perform very well on various languages and on cross-lingual transfer between them (e.g., Kondratyuk and Straka, 2019; Pasini et al., 2021). Due to this success, there has been a great deal of interest in uncovering what these models learn from the multilingual pretraining signal. However, these works analyze a single model artifact: the final training checkpoint at which the model is considered to be converged. Recent work has also studied monolingual models by expanding the analysis to multiple pretraining checkpoints to see how model knowledge changes across time (Liu et al., 2021).

We analyze multilingual training checkpoints throughout the pretraining process in order to identify when multilingual models obtain their in-language and cross-lingual abilities. The case of multilingual language models is particularly interesting, as the model learns both to capture individual languages and to transfer between them just from unbalanced multitask language modeling for each language.

Specifically, we retrain a popular multilingual model, XLM-R (Conneau et al., 2020a), and run a suite of linguistic tasks covering 59 languages on checkpoints from across the pretraining process.[2] This suite evaluates different syntactic and semantic skills in both monolingual and cross-lingual transfer settings. While our analysis primarily focuses on the knowledge captured in model output representations over time, we also consider how the performance of internal layers changes during pretraining for a subset of tasks.

Our analysis uncovers several insights into multilingual knowledge acquisition. First, while the model acquires most in-language linguistic information early on, cross-lingual transfer is learned across the entire pretraining process. Second, the order in which the model acquires linguistic information for each language is generally consistent with monolingual models: lower-level syntax is learned prior to higher-level syntax and then semantics. In comparison, the order in which the

[2]The XLM-R$_{replica}$ checkpoints are available at `https://nlp.cs.washington.edu/xlmr-across-time`.

model learns to transfer linguistic information between specific languages can vary wildly.

Finally, we observe significant degradation of performance for many languages at the final layer of the last, converged model checkpoint. However, lower layers of the network often continue to improve later in pretraining and outperform the final layer, particularly for cross-lingual transfer. These observations indicate that there is not a single time step (or layer) in pretraining that performs the best across all languages and suggest that methods that better balance these tradeoffs could improve multilingual pretraining in the future.

4.2 Analyzing Knowledge Acquisition Throughout Multilingual Pretraining

Our goal is to quantify when information is learned by multilingual models across pretraining. To this end, we reproduce a popular multilingual pretrained model, XLM-R – referred to as XLM-$R_{replica}$ – and retain several training checkpoints (§4.2.1). A suite of linguistic tasks is then run on the various checkpoints (§4.2.2). For a subset of these tasks, we also evaluate at which layer in the network information is captured during pretraining.

Since we want to identify what knowledge is gleaned from the pretraining signal, each task is evaluated without finetuning. The majority of our tasks are tested via *probes*, in which representations are taken from the final layer of the frozen checkpoint and used as input features to a linear model trained on the task of interest (Belinkov et al., 2020). Additional evaluations we consider for the model include an intrinsic evaluation of model learning (BPC) and unsupervised word alignment of model representations. Each of the tasks in our evaluation suite tests the extent to which a training checkpoint captures some form of *linguistic information*, or a specific aspect of linguistic knowledge, and they serve as a proxy for language understanding in the model.

4.2.1 Replicating XLM-R

Analyzing model learning throughout pretraining requires access to intermediate training checkpoints, rather than just the final artifact. We replicate the base version of XLM-R and save a number of checkpoints throughout the training process. Our pretraining setup primarily follows that of the original XLM-R, with the exception that we use a smaller batch size (1024 examples per batch instead of 8192) due to computational constraints. All other hyperparameters remain unchanged.

XLM-R$_{replica}$ consists of the same model architecture as XLM-R$_{base}$, with a total of 270M parameters. We train the model for 1.5 million updates on 64 Nvidia V100 32 GB GPUs using the fairseq library (Ott et al., 2019). Notably, the language sampling alpha for up-weighting less frequent languages is set to $\alpha = 0.7$: this matches the value used for the XLM-R, though it was reported as $\alpha = 0.3$ in the original paper.

XLM-R$_{replica}$ is also trained on the same data as the original model, CC100. This dataset consists of filtered Common Crawl data for 100 languages, with a wide range of data quantities ranging from 0.1 GiB for languages like Xhosa and Scottish Gaelic to over 300 Gib for English. As with XLM-R, we train on CC100 for 1.5M updates and save 39 checkpoints for our analysis, with more frequent checkpoints taken in the earlier portion of training: we save the model every 5k training steps up to the 50k step, and then every 50k steps. Further details about the original data and pretraining scheme can be found in Conneau et al. (2020a).

Comparing XLM-R$_{base}$ and XLM-R$_{replica}$ We compare the performance of our retrained XLM-R$_{replica}$ model against the original XLM-R$_{base}$ on a subset of the tasks in our evaluation suite (Table

Task		XLM-R$_{base}$	XLM-R$_{replica}$
In-lang	BPC	0.609*	0.652
	POS	89.65*	87.20
	XNLI	58.08*	55.73
X-lang	POS	66.01*	64.94
	XNLI	53.26	53.77*

Table 4.1: Average performance across languages of XLM-R$_{base}$ and the final checkpoint of XLM-R$_{replica}$.

Task	Setup	Num. Langs (Pairs) In-lang.	X-lang.	Example
BPC	Masked LM	94	–	quick The [MASK] brown fox jumps
POS Tagging	Token Labeling	44	18 → 18	ADJ The quick brown fox jumps
Dependency Arc Pred.	Token Pair Labeling	44	18 → 18	✗ ✓ The quick brown fox jumps
Dependency Arc Class.	Token Pair Labeling	44	18 → 18	amod The quick brown fox jumps
XNLI	Sent. Pair Labeling	15	15 → 15	The quick brown fox jumps The fox is fast Entails
SimAlign	Unsupervised Alignment	–	1 → 6	Le renard brun rapide saute The quick brown fox jumps

Table 4.2: Summary of the linguistic information we probe XLM-R$_{replica}$ for throughout pretraining.

Task	Languages
BPC	af, am, ar, as, az, be, bg, bn, br, bs, ca, cs, cy, da, de, el, en, eo, es, et, eu, fa, fi, fr, fy, ga, gd, gl, gu, ha, he, hi, hr, hu, hy, id, is, it, ja, jv, ka, kk, km, kn, ko, ku, ky, la, lo, lt, lv, mg, mk, ml, mn, mr, ms, my, ne, nl, no, om, or, pa, pl, ps, pt, ro, ru, sa, sd, si, sk, sl, so, sq, sr, su, sv, sw, ta, te, th, tl, tr, ug, uk, ur, uz, vi, xh, yi, zh, zh
UD	af, **ar**, bg, ca, **cs**, cy, da, **de**, el, **en**, **es**, et, eu, fa, **fi**, **fr**, ga, gd, he, **hi**, hr, hu, hy, **is**, **it**, **ja**, **ko**, la, lv, nl, **pl**, **pt**, ro, **ru**, sk, sl, sr, **sv**, **tr**, ug, uk, ur, vi, **zh**
XNLI	**ar**, **bg**, **de**, **el**, **en**, **es**, **fr**, **hi**, **ru**, **sw**, **th**, **tr**, **ur**, **vi**, **zh**

Table 4.3: Table summarizing the languages considered for each task. Languages in bold are also used for the cross-lingual setting of the task. UD covers all of the languages used for POS tagging, dependency arc prediction, and dependency arc classification.

4.1). We find that on average, the original XLM-R model achieves better BPC than the replicated model; this is likely due to the decrease in batch size while retraining the model. The replica model also performs slightly worse than the original on in-language tasks but comparably cross-lingually (and outperforms the original model on cross-lingual XNLI).

4.2.2 Linguistic Information Tasks

The analysis suite covers different types of syntactic knowledge, semantics in the form of natural language inference, and word alignment (Table 4.2). These tasks evaluate both in-language linguistics as well as cross-lingual transfer with a wide variety of languages and language pairs. Most tasks (POS tagging, dependency structure tasks, and XNLI) are evaluated with accuracy; the MLM

evaluation is scored on BPC, and SimAlign is evaluated on F1 performance. Table 4.3 presents the languages included in each probing task. We filter the Romanized versions of languages from the CC100 dataset, leaving us with up to 94 for evaluation.

MLM Bits per Character (BPC) As an intrinsic measure of model performance, we consider the bits per character (BPC) on each training language of the underlying MLM. For a sequence **s**, BPC(**s**) is the (average) negative log-likelihood (NLL) of the sequence under the model normalized by the number of characters per token; lower is better for this metric. These numbers are often not reported for individual languages or across time for multilingual models, making it unclear how well the model captures each language on the pretraining task. We evaluate BPC on the validation split of CC100.

Part-of-Speech (POS) Tagging We probe XLM-R$_{replica}$ with a linear model mapping each word's representation to its corresponding POS tag; words that are split into multiple subword tokens in the input are represented by the average of their subword representations. The probes are trained using the Universal Dependencies (UD) treebanks for each language (Nivre et al., 2020). For cross-lingual transfer, we evaluate using Parallel Universal Dependencies (PUD; Zeman et al., 2017), a set of parallel test treebanks, to control for any differences in the evaluation data.

Dependency Structure We evaluate syntactic dependency structure knowledge with two pairwise probing tasks: *arc prediction*, in which the probe is trained to identify pairs of words that are linked with a dependency arc; and *arc classification*, where the probe labels a pair of words with their corresponding dependency relation. The two word-level representations r_1 and r_2 are formatted as a single concatenated input vector $[r_1; r_2; r_1 \odot r_2]$, following the method presented in Chapter 2. This combined representation is then used as the input to a linear model that labels the word pair. Probes for both dependency tasks are trained and evaluated with the same set of UD treebanks as POS tagging.

XNLI We also consider model knowledge of natural language inference (NLI), where the probe is trained to determine whether a pair of sentences entail, contradict, or are unrelated to each other. Given two sentences, we obtain their respective representation r_1 and r_2 by averaging all representations in the sentence, and train the probe on the concatenated representation $[r_1; r_2; r_1 \odot r_2]$. We train and evaluate the probes with the XNLI dataset (Conneau et al., 2018); for training data outside of English, we use the translated data provided by Singh et al. (2019).

Word Alignment In the layer-wise evaluation (§4.5), we evaluate how well the model's internal representations are aligned using SimAlign (Sabet et al., 2020), an unsupervised algorithm for aligning bitext at the word level using multilingual representations. We evaluate the XLM-R$_{replica}$ training checkpoints with SimAlign on manually annotated reference alignments for the following language pairs: EN-CS (Mareček, 2008), EN-DE[3], EN-FA (Tavakoli and Faili, 2014), EN-FR (WPT2003, Och and Ney, 2000), EN-HI[4], and EN-RO[4].

4.2.3 Linguistic Probe Experimental Details

Each evaluation is run on the frozen parameters of a training checkpoint of XLM-R$_{replica}$. All representations are taken from the final (12th) layer of the encoder, except for the experiments presented in §4.5, which consider the performance of different layers within the model over time.

For the linguistic information tasks involving probing, each probe consists of a single linear layer, trained with a batch size of 256 for 50 epochs with early stopping performed on the validation set. The probes, therefore, consist of a limited number of parameters $m * l$, where $m = 768$ is the output dimension of the model and l is the size of the task label set. Following Liu et al. (2019a), the probes are optimized with a learning rate of 1e-3. Each probe is trained on a single Nvidia V100 16GB GPU and takes between <1 minute and 6 minutes to train (depending on dataset size, which varies by language and task). The reported results for each probe are the averaged performance across five runs. For SimAlign, we use the default settings provided in the SimAlign

[3]Gold alignments on EuroParl (Koehn, 2005), http://www-i6.informatik.rwth-aachen.de/goldAlignment/
[4] WPT2005, http://web.eecs.umich.edu/ mihalcea/wpt05/

implementation.[5] We report word-level alignment performance (instead of sub-word alignment) using the itermax alignment algorithm.

Generally, the probe is trained and evaluated on the same language. However, for the cross-lingual experiments in §4.4 and §4.5.2, we instead train the probe on one language (the *source* language) and use that *source* probe to evaluate the model on other, *target* languages.

4.3 In-language Learning Throughout Pretraining

We first consider the in-language, or monolingual, performance of XLM-R$_{replica}$ on different types of linguistic information across pretraining. We find that in-language linguistics is learned (very) early in pretraining and is acquired in a consistent order, with lower-level syntactic information learned before more complex syntax and semantics. Additionally, the final checkpoint of XLM-R$_{replica}$ often experiences performance degradation compared to the best checkpoint for a language, suggesting that the model is forgetting information for a number of languages by the end of pretraining.

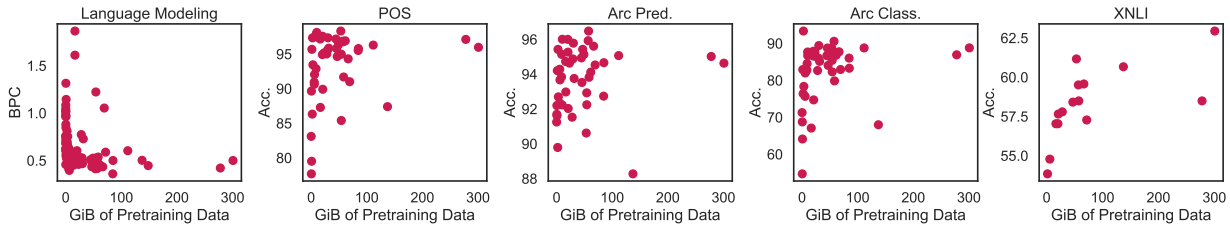


Figure 4.1: Best in-language performance of XLM-R$_{replica}$ on various tasks and languages across all checkpoints.

4.3.1 Monolingual Performance for Different Languages

Figure 4.1 presents the overall best performance of the model across time on the considered tasks and languages. We observe a large amount of variance in performance on each task. Across languages, the performance of XLM-R$_{replica}$ ranges between 1.86 and 0.36 BPC for language modeling, 88.3% and 96.5% accuracy for dependency arc prediction, 77.67% and 98.3% accuracy

[5] https://github.com/cisnlp/simalign

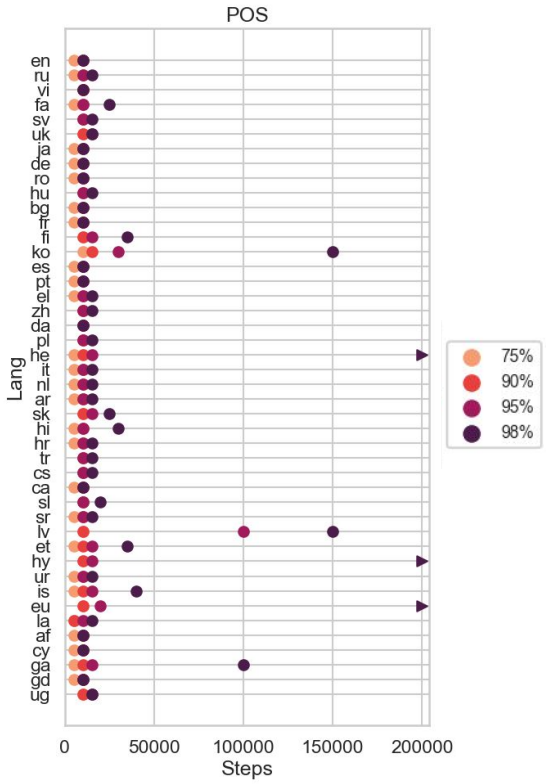


Figure 4.2: Learning progress of XLM-R$_{replica}$ on POS tagging, up to 200k training steps. Each point represents the step where the model achieves x% of it's best overall performance on that task.

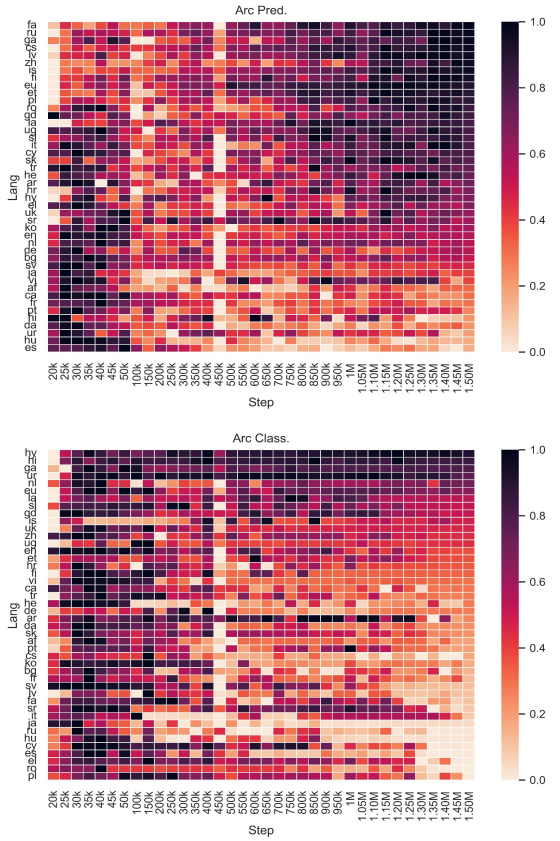


Figure 4.3: Heatmap of relative performance over time for dependency arc prediction and classification. Languages are ordered by performance degradation in the final training checkpoint.

for POS tagging, 54.7% and 93.3% accuracy for arc classification, and 53.8% and 62.9% accuracy for XNLI. Overall, these results confirm previous findings that multilingual model performance varies greatly on different languages.

4.3.2 When Does XLM-R Learn Linguistic Information?

Figure 4.2 shows the step at which XLM-R$_{replica}$ reaches different percentages of its overall best performance.

Monolingual linguistics is acquired early in pretraining We find that XLM-R$_{replica}$ acquires the majority of in-language linguistic information early in training. However, the average time step

for acquisition varies across tasks. For dependency arc prediction, all languages achieve 98% or more of total performance by 20k training steps (out of 1.5M total updates). In contrast, XNLI is learned later with the majority of the languages achieving 98% of the overall performance after 100k training updates. This order of acquisition is in line with monolingual English models, which have also been found to learn syntactic information before higher-level semantics (Liu et al., 2021).

We also observe that this order of acquisition is often maintained within individual languages. 12 out of 13 of the languages shared across all tasks reach 98% of the best performance consistently in the order of POS tagging and arc prediction (which are typically learned within one checkpoint of each other), arc classification, and XNLI.

Model behavior later in pretraining varies across languages For some languages and tasks, XLM-R$_{replica}$ never achieves good absolute performance (Figure 4.1). For others, the performance of XLM-R$_{replica}$ decreases later in pretraining, leading the converged model to have degraded performance on those tasks and languages (Figure 4.3). We hypothesize that this is another aspect of the "curse of multilinguality," where some languages are more poorly captured in multilingual models due to limited capacity (Conneau et al., 2020a; Wang et al., 2020), arising during the training process. We also find that the ranking of languages by performance degradation is not correlated across tasks. This suggests the phenomenon is not limited to a subset of low-resource languages and can affect any language learned by the model.

More generally, these trends demonstrate that the best model state varies across languages and tasks. Since BPC continues to improve on all individual training languages throughout pretraining, the results also indicate that performance on the pretraining task is not directly tied to performance on the linguistic probes. This is somewhat surprising, given the general assumption that better pretraining task performance corresponds to better downstream task performance.

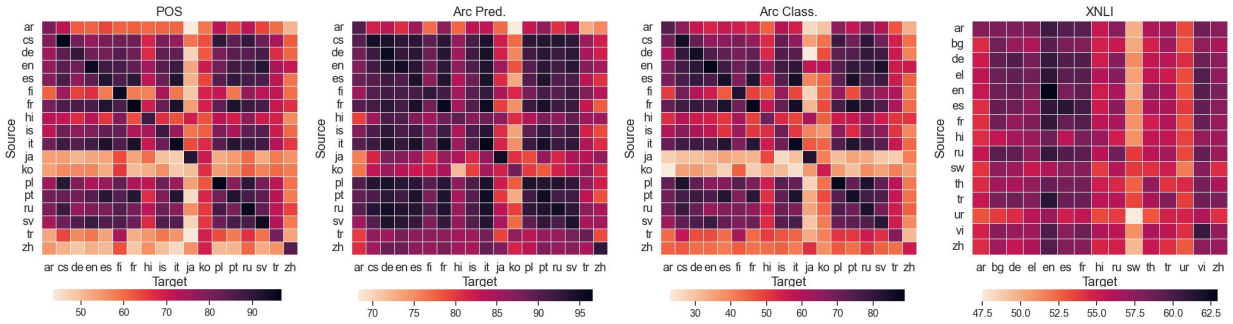


Figure 4.4: Overall performance of XLM-R$_{replica}$ on each analysis task when transferring from various source to target languages.

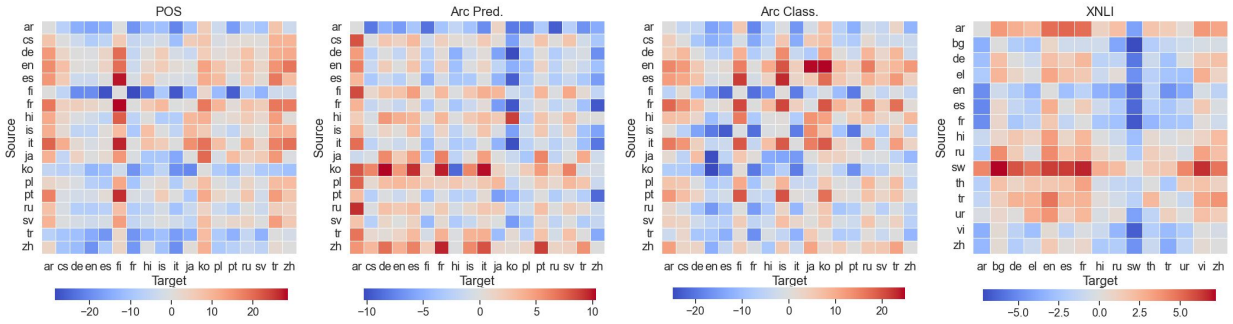


Figure 4.5: Heatmap of the asymmetry of cross-lingual transfer in XLM-R$_{replica}$. Each cell shows the difference in performance between language pairs ($l_1 \rightarrow l_2$) and ($l_2 \rightarrow l_1$).

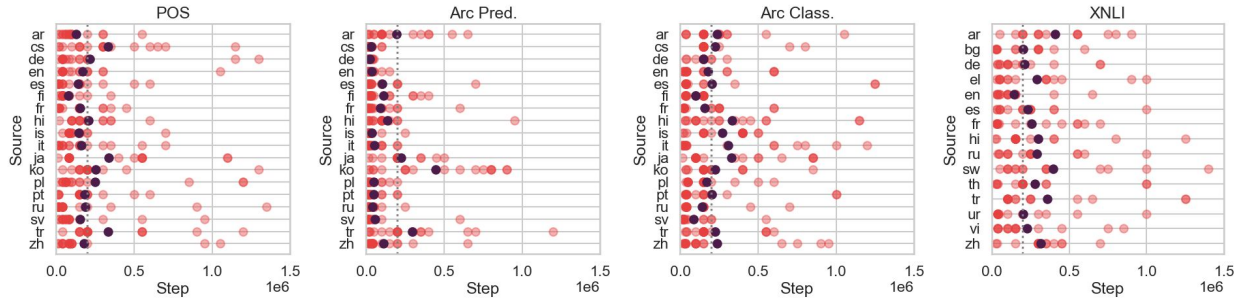


Figure 4.6: Cross-lingual learning progress of XLM-R$_{replica}$ across pretraining. Each red point represents the step to 98% of the best performance for a language pair; the purple represents the mean 98% transfer step for the source language.

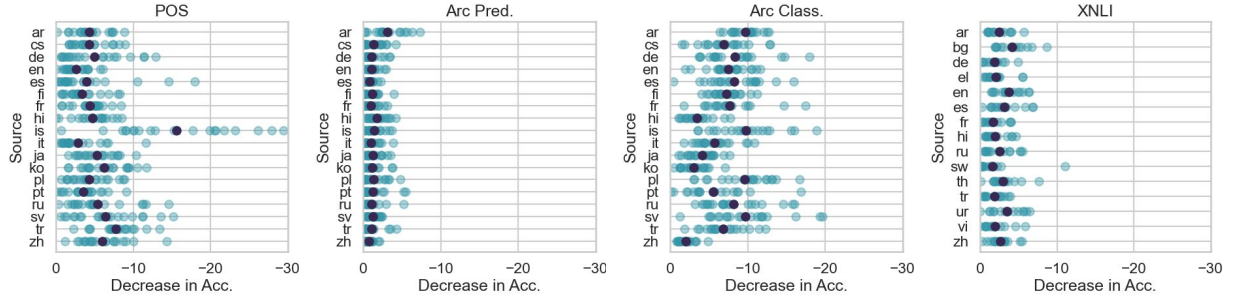


Figure 4.7: Degradation of cross-lingual transfer performance of XLM-R$_{replica}$ across pretraining. Each blue point represents the change in performance from the overall best step to the final model checkpoint for a language pair; the navy represents the mean decrease for the source language.

4.4 Cross-lingual Transfer Throughout Pretraining

Another question of interest is: when do multilingual models learn to transfer between languages? We find that cross-lingual transfer is acquired later in pretraining than monolingual linguistics and that the step at which XLM-R$_{replica}$ learns to transfer a specific language pair varies greatly. Furthermore, though the order in which XLM-R$_{replica}$ learns to transfer different linguistic information across languages is — on average — consistent with in-language results, the order in which the model learns to transfer across specific language pairs for different tasks is much more inconsistent.

4.4.1 Overall Transfer Across Language Pairs

Which languages transfer well? Figure 4.4 shows cross-lingual transfer between different language pairs; most source languages perform well in-language (the diagonal). We observe that some tasks, specifically dependency arc prediction, are easier to transfer between languages than others; however, across the three tasks with shared language pairs (POS tagging, arc prediction, and arc classification) we see similar behavior in the extent to which each language transfers to others. For example, English and Italian both transfer well to most of the target languages. However, other languages are isolated and do not transfer well into or out of other languages, even though in some cases the model achieves good in-language performance.

On XNLI, there is more variation in in-language performance than is observed for syntactic tasks. This stems from a more general trend that some languages appear to be easier to transfer into than others, leading to the observed performance consistency within columns. For example, English appears to be particularly easy for XLM-R$_{replica}$ to transfer into, with 12 out of the 14 source languages performing as well or better on English as in-language.

Cross-lingual transfer is asymmetric We also find that language transfer is asymmetric within language pairs (Figure 4.5). There are different transfer patterns between dependency arc prediction and the other syntactic tasks: for example, we see that Korean is worse relatively as a source language than as the target for POS tagging and arc classification, but performs better when

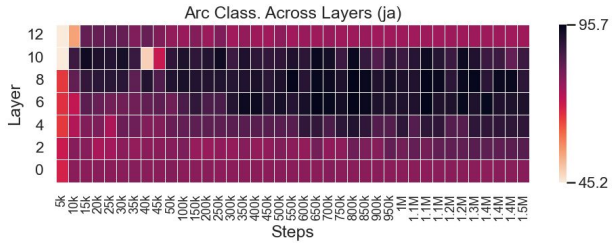
transferring to other languages in arc prediction. However, other languages such as Arabic have similar trends across the syntactic tasks. On XNLI, we find that Swahili and Arabic are the most difficult languages to transfer into, though they transfer to other languages reasonably well.

These results expand on observations in Turc et al. (2021) and emphasize that the choice of source language has a large effect on cross-lingual performance in the target. However, there are factors in play in addition to linguistic similarity causing this behavior, leading to asymmetric transfer in a language pair.

4.4.2 When is Cross-lingual Transfer Learned During Pretraining?

We next consider when during pretraining XLM-R$_{replica}$ learns to transfer between languages (Figure 4.6; the dotted line indicates the 200k step cutoff used in Figure 4.2 for comparison). Unlike the case of monolingual performance, the step at which the model acquires most cross-lingual signal (98%) varies greatly across language pairs. We also find that (similar to the in-language setting) higher-level linguistics transfer later in pretraining than lower-level ones: the average step for a language pair to achieve 98% of overall performance occurs at 115k for dependency arc prediction, 200k for POS tagging, 209k for dependency arc classification, and 274k for XNLI. In contrast, when the model learns to transfer different linguistic information between two specific languages can vary wildly: only approximately 21% of the language pairs shared across the four tasks transfer in the expected order.

We also investigate the amount to which the cross-lingual abilities of XLM-R$_{replica}$ decrease over time (Figure 4.7). Similarly to in-language behavior, we find that the model exhibits notable performance degradation for some language pairs (in particular on POS tagging and dependency arc classification), and the extent of forgetting can vary wildly across target languages for a given source language.



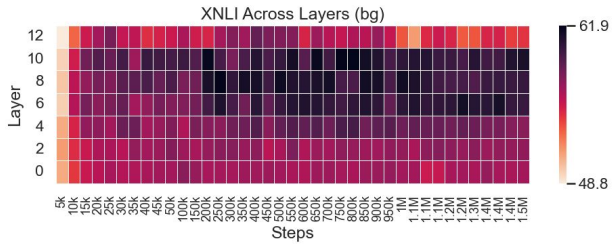
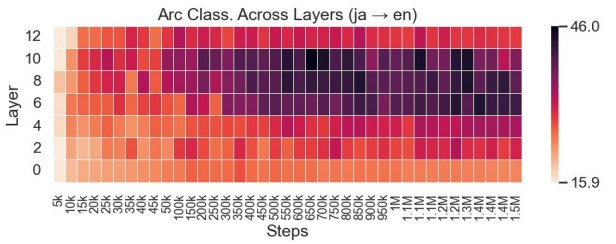


Figure 4.8: Heatmap of XLM-R$_{replica}$ performance for Japanese arc classification and Bulgarian XNLI.



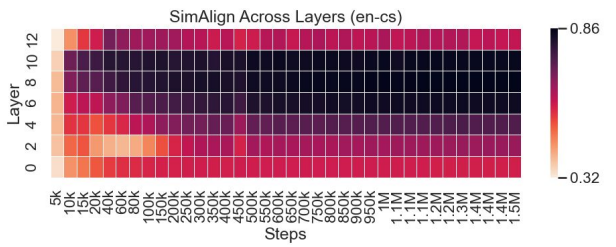


Figure 4.9: Heatmap of XLM-R$_{replica}$ cross-lingual performance by layer for arc classification (JA → EN) and SimAlign (EN-CS).

4.5 Layer-wise Learning Throughout Pretraining

In the experiments above we show that in many cases the final layer of XLM-R$_{replica}$ forgets information by the end of pretraining. Motivated by this, we investigate whether this information is retained in a different part of the network by probing how information changes *across* layers during pretraining. We find a surprising trend in how the best-performing layer changes over time: the model acquires knowledge in higher layers early on, which then propagates to and improves in the lower layers later in pretraining.

4.5.1 In-language Knowledge Across Layers

We first look at the layer-wise performance of XLM-R$_{replica}$ on a subset of languages for dependency arc classification (CS, EN, HI, and JA) and XNLI (BG, EN, HI, and ZH) (Figure 4.8). We find that the last layer is often not the best one for each task, with lower layers often outperforming the final one. On average, the best internal layer state outperforms the final layer of XLM-R$_{replica}$ by 7.59 accuracy points on arc classification and 2.93 points on XNLI.

We also observe a trend of lower layers acquiring knowledge later in training than the final one. To investigate this, we calculate the expected best layer (i.e., the average layer weighted by

performance) at each checkpoint and find that it decreases over time, by up to 2.79 layers for arc classification and 2.49 layers for XNLI, indicating that though the final layer quickly fits to the forms of in-language information we test for, this information then shifts to lower layers in the network over time.

4.5.2 Cross-lingual Knowledge Across Layers

Next, we consider how cross-lingual transfer skills are captured across layers during pretraining. Every other XLM-R$_{replica}$ layer is evaluated on the subsets of languages for arc classification and XNLI in §4.5.1. We also use SimALign to test how well word representations at these layers align from English to {CS, DE, FA, FR, HI, RO}.We observe similar trends with respect to layer performance over time to the in-language results (Figure 4.9). Specifically, we observe an average decrease in the expected layer of 1.10 (ranging from 0.67 to 2.20) on arc classification, 1.02 (ranging from 0.37 to 2.01) on XNLI, and 1.66 (ranging from 0.83 to 2.41) on SimAlign.

We also observe that while most layers perform relatively well in-language performance, the lowest layers of XLM-R$_{replica}$ (layers 0-4) often perform much worse than the middle and final layers for cross-lingual transfer throughout the pretraining process – for example, in the case of Japanese to English on arc classification. We hypothesize that this is due to better alignment across languages in later layers, similar to the findings in Muller et al. (2021).

4.6 What Factors Affect Multilingual Learning?

This section presents extended results analyzing the correlations between different factors and the in-language and cross-lingual learning exhibited by XLM-R$_{replica}$.

4.6.1 In-language Correlation Study

We consider whether the following factors correlate with various measures of model learning (Table 4.4): *pretraining data*, the amount of text in the CC100 corpus for each language; *task data*, the

Variable	Factors	Spearman (ρ)				
		BPC	POS	Arc Pred.	Arc Class.	XNLI
Task Perf.	Pretraining Data	-0.597**	0.258	0.267	0.411*	0.767**
	Task Data	-0.597**	0.462*	0.276	0.527**	
	Lang Sim.	0.427**	-0.315*	-0.170	-0.427*	-0.779**
Steps to 95%	Pretraining Data	0.135	-0.290	-0.193	-0.301*	-0.239
	Task Data	0.135	-0.065	-0.260	-0.209	
	Lang Sim.	-0.385**	0.156	0.268	0.325*	0.316
Forgetting	Pretraining Data		0.230	0.218	0.437*	0.564*
	Task Data		-0.322*	-0.338*	-0.015	
	Lang Sim.		0.172	-0.158	-0.181	-0.795**

Table 4.4: Correlation study of different factors against measures of in-language knowledge. * $p < 0.05$, ** $p < 0.001$

amount of in-task data used to train each probe; and *language similarity* to English, which is the highest-resource language in the pretraining data. We use the syntactic distances calculated in Malaviya et al. (2017) as our measure of language similarity; these scores are smaller for more similar language pairs.

Overall Performance The amount of pretraining data and in-task training data are strongly correlated with overall task performance for most of the considered tasks; this corroborates similar results from Wu and Dredze (2020). Language similarity with English is also correlated with better in-task performance on all tasks except for dependency arc prediction, suggesting that some form of cross-lingual signal supports in-language performance for linguistically similar languages.

Learning Progress Measures We also consider (1) the step at which XLM-R$_{replica}$ achieves 95% of its best performance for each language and task, which measures how quickly the model obtains a majority of the tested linguistic information, and (2) how much the model *forgets* from the best performance for each language by the final training checkpoint. We find that language similarity to English is strongly correlated with how quickly XLM-R$_{replica}$ converges on BPC and dependency arc classification. This suggests that cross-lingual signal helps the model more quickly learn lower-resource languages on these tasks, in addition to improving overall model performance. However, we observe no strong trends as to what factors affect forgetting across tasks.

Variable	Factors	Spearman (ρ)			
		POS	Arc Pred.	Arc Class.	XNLI
Task Perf.	Src. Pretraining Data	0.113*	0.107	0.117*	0.178*
	Trg. Pretraining Data	0.038	0.144*	0.015	0.625**
	Task Data	0.245**	0.124*	0.129*	
	Lang Sim.	-0.598**	-0.575**	-0.593**	-0.321**
Asymmetry	Src. Pretraining Data	0.116*	-0.045	0.140*	-0.423*
	Trg. Pretraining Data	-0.116*	0.045	-0.140*	0.423*
	Task Data	0.123*	-0.016	-0.077	
Steps to 95%	Src. Pretraining Data	-0.290**	-0.023	-0.132*	-0.195*
	Trg. Pretraining Data	-0.123*	-0.066	-0.106	-0.057
	Task Data	0.073	-0.057	0.115*	
	Lang Sim.	0.475**	0.518**	0.492**	0.076
Forgetting	Src. Pretraining Data	-0.208**	-0.123*	0.000	0.137*
	Trg. Pretraining Data	0.042	0.015	0.122*	-0.079
	Task Data	0.009	-0.004	0.078	
	Lang Sim.	0.165*	0.186*	-0.025	0.164*

Table 4.5: Correlation study of different factors against measures of cross-lingual transfer. * p < 0.05, ** p < 0.001

4.6.2 Cross-lingual Correlation Study

Table 4.5 presents a correlation study of different measures for cross-lingual transfer in XLM-R$_{replica}$. We consider the effect of source and target pretraining data quantity, the amount of in-task training data (in the source language), and the similarity between the source and target language on the following transfer measures: overall task performance, asymmetry in transfer (the difference in model performance on $l_1 \rightarrow l_2$ compared to $l_2 \rightarrow l_1$), the step at which the model achieves 95% or more of overall performance on the language pair, and forgetting – the (relative) degradation of overall performance in the final model checkpoint.

Correlations of Transfer with Language Factors For overall cross-lingual performance, we observe that language similarity is highly correlated with task performance for all tasks and is similarly correlated with speed of acquisition (the step to 95% of overall performance) for three of the four considered tasks. This is in line with prior work that has also identified language similarity as a strong indicator of cross-lingual performance (Pires et al., 2019). However, all considered factors are less correlated with the other measures of knowledge acquisition, such as the asymmetry

of transfer and the forgetting of cross-lingual knowledge; this suggests that there could be other factors that explain these phenomena.

Interactions Between Learning Measures We also consider the correlations between the different measures of model performance on cross-lingual transfer. For example, overall transfer performance is strongly correlated (p«0.001) with earlier acquisition (step to 95% of overall performance) for all syntactic tasks: $\rho = -0.50$ for both POS tagging and dependency arc prediction and -0.55 for arc classification. To a lesser extent, overall transfer performance and model forgetting are negatively correlated, ranging from $\rho = -0.13$ to -0.42 across considered tasks. This indicates that XLM-R$_{replica}$ forgets less of the learned cross-lingual signal for better-performing language pairs, at the expense of already less successful ones.

4.7 Discussion

In this chapter, we probe training checkpoints across time to analyze the training dynamics of the XLM-R pretraining process. We find that although the model learns in-language linguistic information early in training – similar to findings on monolingual models – cross-lingual transfer is obtained all throughout the pretraining process.

Furthermore, the order in which linguistic information is acquired by the model is generally consistent, with lower-level syntax acquired before semantics. However, we observe that for individual language pairs this order can vary wildly, and our statistical analyses demonstrate that model learning speed and overall performance on specific languages (and pairs) are difficult to predict from language-specific factors.

We also observe that the final model artifact of XLM-R$_{replica}$ performs often significantly worse than earlier training checkpoints on many languages and tasks. However, layer-wise analysis of the model shows that linguistic information shifts lower in the network during pretraining, with lower layers eventually outperforming the final layer. Altogether, these findings provide a better understanding of multilingual training dynamics that can inform future pretraining approaches.

Chapter 5

Cross-lingual Expert Language Models

Despite their popularity in NLP beyond English, multilingual language models often underperform monolingual ones due to inter-language competition for model parameters. We propose Cross-lingual Expert Language Models (X-ELM), which mitigate this competition by independently training language models on subsets of the multilingual corpus. This process specializes X-ELMs to different languages while remaining effective as a multilingual ensemble. Our experiments show that when given the same compute budget, X-ELM outperforms jointly trained multilingual models across all considered languages and that these gains transfer to downstream tasks. X-ELM provides additional benefits over performance improvements: new experts can be iteratively added, adapting X-ELM to new languages without catastrophic forgetting. Furthermore, training is asynchronous, reducing the hardware requirements for multilingual training and democratizing multilingual modeling.[1]

5.1 Introduction

Massively multilingual language models (LMs), which are trained on terabytes of text in a hundred or more languages, underlie almost all multilingual and cross-lingual NLP applications (Scao et al.,

[1]This chapter presents a summary of, and includes materials originally published in, Blevins et al. (2024). Appendix C provides more experimental details, analysis, and full numerical results.

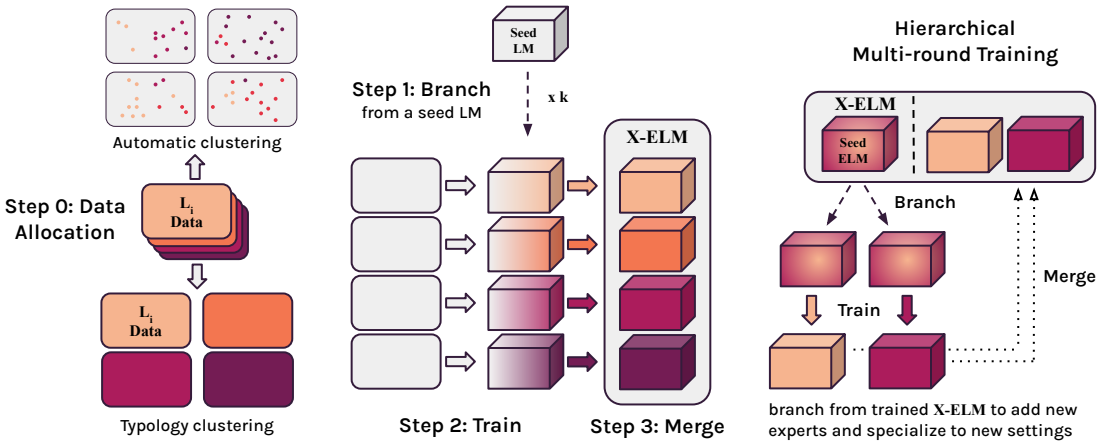


Figure 5.1: Overview of the X-ELM pretraining procedure. **Left**: We partition the multilingual text corpus into k subsets either through *automatic TF-IDF* clustering of documents or through grouping languages by *linguistic typology*. **Center**: Branch-Train-Merge (BTM) pretraining method. We initialize (*branch*) k experts from a seed LM, *train* each expert on a different cluster from the pretraining corpus, and *merge* the experts into a set of X-ELMs. **Right**: Hierarchical Multi-Round (HMR) training procedure (§5.4).

2022; Lin et al., 2022, inter alia). Despite their wide adoption, these models come at a cost: by modeling many languages in a single model, there is inter-language competition for fixed model capacity; this causes performance on individual languages to degrade relative to monolingual models (Conneau et al., 2020a; Chang et al., 2023). Furthermore, this phenomenon (termed the *curse of multilinguality*) can significantly harm low-resource languages (Wu and Dredze, 2020).

In this paper, we address the curse of multilinguality with **Cross**-lingual **E**xpert **L**anguage **M**odels (X-ELM, Figure 5.1), an ensemble of language models initialized from a pretrained multilingual model and each independently trained on a different subset of a multilingual corpus. Our ensemble allows for efficient scaling of model capacity to better represent all the corpus languages. These X-ELMs are trained with x-BTM, a new extension of the Branch-Train-Merge paradigm (BTM; Li et al., 2022; Gururangan et al., 2023, §5.2) to the more heterogenous multilingual setting.

x-BTM improves over existing BTM techniques by introducing (1) a new method for balanced clustering of multilingual data based on typological similarity and (2) Hierarchical Multi-Round training (HMR), an algorithm for efficiently training new experts specialized to unseen languages or other distributions of multilingual data. Once the initial X-ELMs are trained, we dynamically select

experts to perform inference (§5.3.3). We can also efficiently adapt X-ELMs to novel settings with additional rounds of x-BTM on new experts branched from existing X-ELMs (§5.4); this improves the overall X-ELM set without altering the existing experts.

We train X-ELMs on 20 total languages—including adapting to 4 unseen ones–and on up to 21 billion training tokens. Our experiments demonstrate that X-ELMs outperform the dense language models given the same compute budget in every considered experimental setting, with improvements of up to 3.8 perplexity points (§5.6). Furthermore, the perplexity gains observed in X-ELM languages are well-balanced across language resourcedness, and adapting the models to new languages via HMR training significantly outperforms standard language-adaptive pretraining methods. We also show that the language modeling gains of X-ELM hold on downstream task evaluations (§5.7).

Multilingual modeling with X-ELM provides additional benefits over improved performance. Training a set of X-ELMs is more computationally efficient than a comparable dense model; each expert is trained independently, which removes the overhead cost of cross-GPU synchronization (Li et al., 2022) and allows experts to be trained asynchronously in low-compute settings. Similarly, adapting X-ELMs to new languages is more efficient than continued training of a dense LM and does not risk catastrophic forgetting of previously seen languages, as adding a new X-ELM does not change the existing experts. As a result, X-ELMs allow much more efficient modeling than prior multilingual approaches, democratizing work on building and improving multilingual systems.

5.2 Background: Branch-Train-Merge

Multilingual LMs are typically trained in a *dense* manner, where a single set of parameters are updated with every training batch. When training large LMs, the dense training setup calculates gradients on and synchronizes model parameters across many GPUs.[2] This requires all GPUs to be available simultaneously and incurs communication costs that prolong training.

Branch-Train-Merge (BTM; Li et al. 2022) alleviates this cost by dividing the total compute

[2]For example, the XGLM-7.5B model "was trained on 256 A100 GPUs for about 3 weeks" (Lin et al., 2022).

among smaller expert language models that are trained independently on different domains (or subsets of a corpus) and then combined during inference time. While the total number of parameters increases with the number of experts, inference with these models often uses a subset of experts (see §5.3.3), keeping inference costs manageable.

c-BTM (Gururangan et al., 2023) generalizes the above approach with cluster-based representations of domains. Across multiple corpora, they show that (1) the optimal number of experts increases with data and compute and (2) a set of small expert models performs similarly to equivalently sized dense models at vastly reduced FLOP budgets.

Our work extends these studies to the multilingual setting, in which experts are specialized to different languages instead of (primarily) English-language domains. In the multilingual setting, we can also use typological structure to specialize experts, which we show provides additional benefits over automatic data clustering. We also demonstrate that training along the hierarchy of language families in multiple rounds yields further performance benefits.

5.3 Cross-lingual Expert Language Models

Multilingual language models are jointly trained on many different languages (e.g., Lin et al., 2022), despite the well-documented curse of multilinguality that comes from the competition between languages for fixed model capacity (Conneau et al., 2020a; Wang et al., 2020). We propose **Cross-lingual Expert Language Models**, or X-ELMs, to address this performance disparity (Figure 5.1). These experts are trained with **x-BTM**, an extension of the Branch-Train-Merge (BTM) pretraining paradigm (Li et al., 2022; Gururangan et al., 2023): we asynchronously train many expert LMs on subsets of a multilingual corpus in order to specialize them to different sub-distributions of the multilingual space and then merge the experts to perform inference. We hypothesize that this training scheme will alleviate the curse of multilinguality on individual languages while maintaining the cross-lingual properties of dense multilingual LMs.

5.3.1 x-BTM: Sparse Multilingual Training

This section overviews our algorithm for sparse training of multilingual experts.

Step 0: Multilingual Data Allocation As a preprocessing step, we partition the multilingual corpus into k clusters to train each X-ELM. We consider learning TF-IDF clusters as well as a new clustering method that groups documents by language identity and linguistic typology (§5.3.2).

Step 1: Branch A preliminary stage of shared, dense pretraining is important for ensembling expert language models (Li et al., 2022). Therefore, the first step of BTM is to initialize (*branch*) each expert with the parameters from a partially trained model. For this work, we initialize our X-ELMs with an existing multilingual pretrained model, XGLM (Lin et al., 2022).

Step 2: Train After initialization, we assign each expert a data cluster and train for a fixed number of steps with an autoregressive LM objective. Expert training is independent, with no shared parameters between models.

Step 3: Merge We collect the k X-ELMs into a set and perform inference with them. We consider several methods of inference and expert ensembling in §5.3.3.

Steps 1 – 3 describe a single round of x-BTM training. However, we can continue to update the X-ELM set by branching—initializing a new group of experts—from existing models in the ensemble and performing more rounds of x-BTM via the method we propose in §5.4. This allows us to further improve X-ELM by training and adding new experts.

5.3.2 Data Allocation Methods

How we assign data to experts is a key component of training X-ELM, and it is a particularly crucial choice as the data becomes more diverse (i.e., spanning many languages). We consider two methods of data allocation when training our X-ELMs:

Balanced TF-IDF Clustering We partition the multilingual corpus automatically into k components with k-means clustering. First, we encode each document into a word-level TF-IDF representation[3]; we then perform balanced k-means clustering on these representations to obtain approximately balanced subsets of the data on which to train each X-ELM. Further details on the balanced k-means clustering method can be found in Gururangan et al. (2023). This allocation method uses no language information outside of what is inherent in the text (e.g., script, vocabulary).

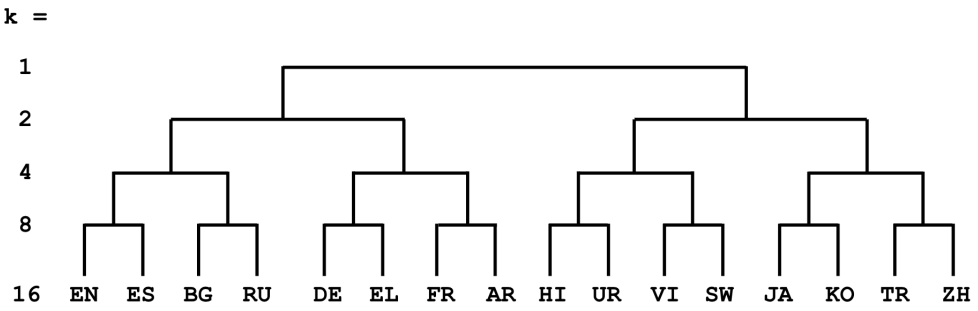


Figure 5.2: Heirachical clustering of languages used to train our X-ELM ensembles.

Linguistic Typology Clustering We also consider segmenting the corpus by language identity.[4] Rather than balancing the amount of data allocated to each cluster in this setting, we instead keep the number of languages per cluster fixed. Specifically, we learn a balanced hierarchical clustering of the languages (Figure 5.2). We build this hierarchy using the language similarity metrics in LANG2VEC (Littell et al., 2017), which represents languages based on linguistic features in resources such as WALS[5] and estimates language similarity with distance in this feature space. We first initialize each cluster with a single language; at each step, we merge each cluster with exactly one other based on the *minimum* distances between the cluster centroids. We then group languages according to the resulting hierarchy and the desired number of experts. When the number of languages equals the number of experts, typological clustering results in monolingual training, where every language is assigned a separate expert.

[3]Data tokenization is independent of the downstream model. Here, we use the sklearn text-vectorizer tokenizer.
[4]This requires knowledge of the language of each document. We use the language tags provided with mC4.
[5]World Atlas of Language Structures, `https://wals.info/`

Comparing the Clustering Techniques Figure 5.3 shows the difference in language distributions between the *TF-IDF* and *Linguistic Typology* clusters. While *TF-IDF* allows language data to spread across experts, we find that, in practice, the distributions remain relatively sparse. The main exception is at $k = 16$, when the highest-resourced languages in the data (e.g., English or Russian) are split across clusters due to the constraint that balances the amount of data per cluster.

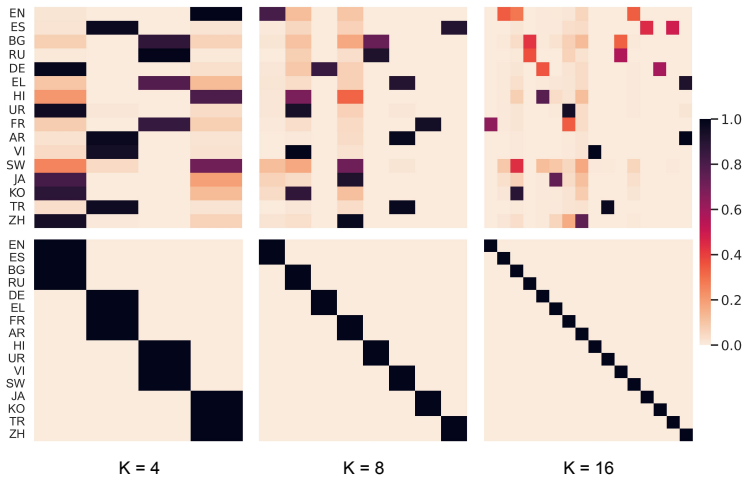


Figure 5.3: Percentage of language data assigned to different experts with TF-IDF (top row) and Typ. (bottom row) clustering. For Typ. clustering, each language is assigned entirely to a single expert.

5.3.3 Inference with X-ELMs

We evaluate a number of different methods for performing inference with X-ELMs:

Top-1 Expert This method performs inference with a single expert chosen prior to evaluation; therefore, it incurs the same inference cost as the dense baselines. When evaluating the *Typology* experts on a particular language ℓ, we choose the expert that included ℓ in the set of languages on which they continued pretraining. Similarly, when evaluating *TF-IDF*, we choose the X-ELM trained on the highest percentage of ℓ's data.

Ensembling TF-IDF Experts We also consider ensembling TF-IDF experts by adapting the c-BTM ensemble routing method. Here, we calculate ensembling alphas, or weights, over these experts for *each* evaluation step based on the proceeding context's TF-IDF distance from the experts' k-means centroids. These weights are then used to ensemble the output probabilities from each

expert.

More specifically, given a probability from each expert LM $p_e(x_t|x_{<t})$ and the corresponding ensemble weight $\alpha_e = p(e|x_{<t}) \propto \exp(-\text{dist}(x_{<t}, c_e)^2/T)$, the probability of the ensemble $p_E(x_t|x_{<t}) = \sum_{e \in E} \alpha_e \cdot p_e(x_t|x_{<t})$. Here, $\text{dist}(x_{<t}, c_e)$ is obtained by embedding $x_{<t}$ with the learned TF-IDF vectorizer and calculating the Euclidean distance from c_e (the centroid over the data representations allocated to expert e), and T is a temperature parameter over the ensemble weight distribution. Further details and motivation for this setting are provided in Gururangan et al. (2023).

Ensembling X-ELM outputs increases the cost of inference relative to the dense model or top-1 inference. However, it can potentially better fit different subsets of data in a diverse evaluation set. We also do not assume we know the identity of each example when ensembling, which makes this approach more flexible than the top-1 setting. In most cases, we ensemble all k experts; however, we can also reduce computational costs by *sparsifying* the ensemble weights and only activating the m ($< k$) experts that most contribute to an example: $p_E(x_t|x_{<t}) = \sum_{e \in E} \alpha_e \cdot p_e(x_t|x_{<t}) : \alpha_e \in \text{top-m}(\alpha_{\mathbf{E}})$.

5.4 Hierarchical Multi-Round Training

We previously described a single round of training for X-ELM (§5.3.1). However, BTM can also be used repeatedly to train new experts seeded with those learned in a prior round. The multilingual setting provides a natural extension of multi-round training that leverages typological structure when initializing new experts.

We propose **Hierarchical Multi-Round** (**HMR**) pretraining (Figure 5.1), which uses the learned typological tree structure from *Linguistic Typology* clustering to iteratively train more specific X-ELMs. Specifically, given an expert model x trained on a cluster of languages L, we initialize a new set of experts $X' = x'_1, x'_2, ..., x'_n$ with the parent expert x. Each new expert in X' is then further trained on a different sub-cluster $\ell' \subset L$.

HMR pretraining gives multiple benefits over single-round BTM. In particular, HMR training

Language	mC4† Size (%)	XGLM Size
AR (Arabic)	243.14 (4.1%)	64.34
BG (Bulgarian)	109.3 (1.9%)	61.10
DE (German)	615.59 (10.4%)	369.30
EL (Greek)	193.63 (3.3%)	180.37
EN (English)	877.43 (14.8%)	3,324.45
ES (Spanish)	723.17 (12.2%)	363.83
FR (French)	506.74 (8.6%)	303.76
HI (Hindi)	125.44 (2.1%)	26.63
JA (Japanese)	764.71 (12.9%)	293.39
KO (Korean)	91.29 (1.5%)	79.08
RU (Russian)	957.02 (16.2%)	1,007.38
SW (Swahili)	3.06 (0.05%)	3.19
TR (Turkish)	248.07 (4.2%)	51.51
UR (Urdu)	10.15 (0.2%)	7.77
VI (Vietnamese)	296.65 (5.0%)	50.45
ZH (Chinese)	143.68 (2.4%)	485.32
AZ (Azerbaijani)	15.23 (–)	–
HE (Hebrew)	67.14 (–)	–
PL (Polish)	393.85 (–)	–
SV (Swedish)	154.54 (–)	–

Table 5.1: The frequencies and relative percentages of different languages in our training corpus (†an mC4 subsample) and in the XGLM corpus, CC100-XL (Lin et al., 2022). Sizes are in gigabytes (GiB). EN, ES, FR, and RU are downsampled to 1,024 shards for mC4.

saves compute and more easily adapts our X-ELMs to new settings. A specific application of this is adding new languages to the model: while updating dense multilingual LMs with new languages is difficult and can lead to catastrophic forgetting of existing languages (Winata et al., 2023), hierarchically training an expert on a new language adds it to the X-ELM set without altering the existing information in other experts. We further consider this use case for HMR training in §5.6.3.

5.5 Experimental Design

We present a series of experiments to test whether the X-ELM pretraining paradigm remedies the decrease in individual language performance observed in dense multilingual models.

5.5.1 Pretraining Data and Languages

We train our X-ELMs on mC4, an open-source, multilingual pretraining corpus derived from CommonCrawl (Xue et al., 2021).[6] mC4 provides language tags for each document in the corpus, which were automatically assigned with cld3[7] when the dataset was constructed; we use these language tags during typological clustering (§5.3.2). We focus our experiments on the 16 highest-resourced languages out of the 30 languages on which the seed LM, XGLM-1.7B, was trained. For languages with significantly more data than the others (e.g., English), we subsample their data to the first 1,024 shards. Table 5.1 summarizes the languages we use, as well as their frequencies in the original XGLM pretraining dataset and in our sub-sampled mC4 corpus; the final group of languages is used in §5.6.3.

5.5.2 Pretraining Settings

Each expert in the X-ELM experiments is a 1.7B parameter model with the same architecture as the 1.7B XGLM model (Lin et al., 2022), and they are initialized with XGLM's weights in the initial round of BTM training. Unless otherwise stated, we keep the training parameters from the original XGLM training procedure. We train the experts for a fixed number of training steps and control for the number of tokens seen during training. This ensures that all experts in a setting see the same amount of data (and undergo the same number of training updates) and that experiments across different expert set sizes but under the same training budget are comparable. For most

# Tokens	k	# GPUs	# updates	grad acc.
10.5 B	1	8	20,000	32
	4	4	20,000	16
	8	4	20,000	8
	16	2	20,000	8
21.0 B	1	8	40,000	32
	4	4	40,000	16
	8	4	40,000	8
	16	2	40,000	8

Table 5.2: Overview of the compute budget and resources used for different X-ELM experiments. **k** is the number of experts, **# GPUs** indicates the number of GPUs used to train each expert, and **grad acc.** gives the number of gradient accumulation steps used.

[6] While one could also continue pretraining with the same corpus that the seed LM was trained on, the pretraining data for XGLM is not publicly available.

[7] https://github.com/google/cld3

experiments, we use a shared budget of 10.5B tokens and 20,000 training steps; where indicated, we increase this to 21.0B tokens (40,000 steps) to test the effect of further training.

Table 5.2 presents the compute allocated to each expert and setting at different compute budgets of the X-ELM experiments. The per-model instance batch size (**bsz**) for all experiments is 2, and each training example had a sequence length (**seq. len**) of 2048. The total token budget (**# Tokens**) is the product of (k, # GPUs, # updates, grad acc., bsz, seq. len), normalized by the number of GPUs used for model parallelism (2). Experts are trained with a linear decay learning rate schedule; we use a maximum learning rate of $1.5e-4$ after performing preliminary learning rate sweeps.

5.5.3 Perplexity Evaluation

To evaluate the language modeling performance of the X-ELMs, we separately calculate the perplexity on the mC4 validation sets of each pretraining language. For languages with larger evaluation sets, we estimate performance on the first 5,000 validation examples. This perplexity metric is not comparable across languages, as they have different validation sets.

5.6 Language Modeling Experiments

We now test the effectiveness of sparse language modeling in the multilingual setting. First, we determine the optimal number of clusters for our given compute budget and dataset (§5.6.1). We then demonstrate that X-ELMs outperform comparable dense models on seen languages (§5.6.2) and more effectively adapt to new, unseen languages (§5.6.3). Finally, we examine the effect of sparse training on *forgetting* previously-held knowledge of languages in specific X-ELM experts (§5.6.4).

5.6.1 Choosing the Number of X-ELMs

We first consider which choice of k clusters gives the best multilingual language modeling performance. Figure 5.4 compares the choice of $k = 1, 4, 8, 16$ X-ELMs when trained on 10.5B tokens.[8]

[8]The $k = 16$ setting is equivalent to training monolingual experts for every language.

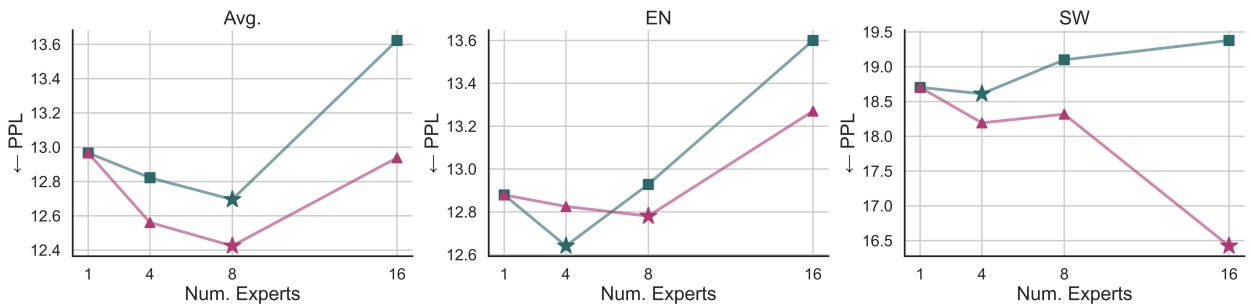


Figure 5.4: Average and language-specific (EN and SW) perplexities across expert counts (k) when clustering with TF-IDF$_{top1}$ (square) and Linguistic Typology (triangle). The best k for each setting is marked with a star.

$k = 8$ is the best-performing setting on 75% of languages when clustering with TF-IDF and for 15 of the 16 pretraining languages when clustering by language similarity. Furthermore, typological clustering consistently outperforms TF-IDF.

These experiments indicate that, for the budget we evaluate, **the best overall X-ELM setting is bilingual models (k=8) clustered by language similarity**. This result is surprising, as it is intuitive to assume that simply continuing to pretrain each expert on a single language (i.e., the $k = 16$ setting) would lead to better perplexity. We find that one language, Swahili, does benefit from the monolingual $k = 16$ setting—possibly because Swahili is paired with a distant language (Vietnamese) by the typological clustering process. However, perplexity is higher in the $k = 16$ setting for all other languages, and in some cases even underperforms the dense ($k = 1$) model.

5.6.2 Perplexity Results on Seen Languages

We now examine the performance of X-ELM in the best setting (k = 8) for the sixteen languages seen during BTM training on computational budgets of 10.5B and 21.0B tokens. Table 5.3 presents the perplexities of the TF-IDF clustered X-ELMs as well as the typologically (Typ.) clustered X-ELMs. As baselines, we compare against the original XGLM-1.7B model and a dense model trained on both computational budgets. We find that the best setting, $k = 8$ with typologically clustered experts, improves by 2.97 and 1.20 on average over the seed and dense baseline models and has individual language gains of up to 7.77 and 3.76 over these models, respectively.

Lang.	10.5B Training Tokens					21.0B Training Tokens				
	XGLM	Dense	TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.	Dense	TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.
AR	16.85	15.29	**14.51**	14.56	14.66	14.97	**14.00**	14.05	14.16
BG	11.31	10.44	10.39	10.39	**10.25**	10.34	10.27	10.26	**10.09**
DE	15.53	14.02	**13.41**	13.50	13.42	13.72	**12.95**	13.05	12.97
EL	10.44	9.40	9.20	9.18	**9.17**	9.24	9.03	9.00	**8.98**
EN	14.37	12.88	12.93	**12.73**	12.78	12.69	12.68	**12.47**	12.55
ES	16.02	14.13	13.92	**13.76**	13.99	13.87	13.54	**13.37**	13.69
FR	13.12	11.78	**11.19**	11.28	11.29	11.54	**10.79**	10.88	10.91
HI	18.28	14.28	14.86	14.19	**11.25**	13.68	14.36	13.62	**10.52**
JA	14.57	12.31	11.95	11.95	**11.49**	11.79	11.36	11.37	**10.88**
KO	8.82	7.79	7.72	**7.67**	**7.67**	7.67	7.61	**7.53**	7.54
RU	13.43	12.52	12.14	12.21	**12.08**	12.33	11.83	11.90	**11.74**
SW	19.85	18.70	19.10	18.76	**18.32**	18.61	19.04	18.67	**18.07**
TR	17.81	15.34	14.13	14.28	**13.80**	14.88	13.41	13.58	**13.03**
UR	14.38	13.45	13.40	13.57	**12.60**	13.38	13.26	13.52	**12.20**
VI	13.07	11.39	11.00	10.86	**10.22**	11.09	10.56	10.42	**9.69**
ZH	17.91	13.74	13.28	13.53	**11.98**	13.12	12.61	12.87	**11.24**
Avg.	14.74	12.97	12.70	12.60	**12.19**	12.68	12.33	12.28	**11.77**

Table 5.3: Per-language and average perplexity results for the $k = 8$ X-ELM experiments (original XGLM and $k = 1$ dense model included for comparison). Lower numbers are better. The best setting for each language is bolded per compute budget. *TF-IDF ensemble uses more parameters for inference than other evaluations.

Expert language models outperform dense continued training For most languages (10 of 16), typologically clustered experts are the best-performing setting. For some high-resource languages (EN and ES), ensembling the TF-IDF experts works better than a single expert. However, this inference setting requires more parameters, as it uses all X-ELMs instead of just the single best expert per language. Furthermore, training X-ELMs for longer unsurprisingly outperforms lower compute settings. All of our experimental settings outperform the seed XGLM model; similarly, the experiments with the 21.0B token compute budget perform better than the respective experiment trained with 10.5B tokens.

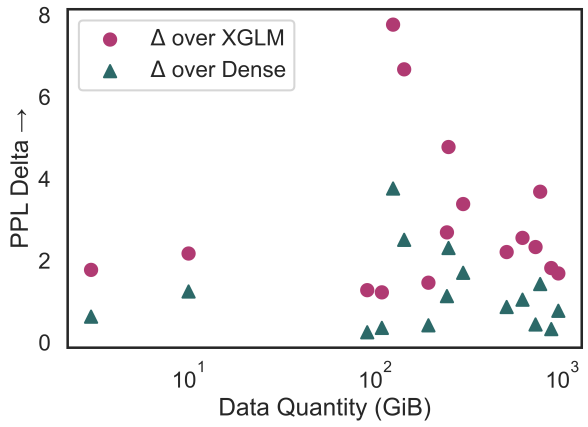


Figure 5.5: Comparison of PPL improvements per language over XGLM-1.7B (circle) and dense baseline (triangle) against the training data quantity (for typologically clustered experts).

X-ELMs improve language modeling on all languages We also show that multilingual language modeling with X-ELMs does not disproportionally benefit languages with more pretraining data (Figure 5.5). Instead, perplexity improvements over both the seed LM and the dense LM baseline *may* slightly favor low-resource languages ($\rho = -0.19, -0.26$, respectively).

Sparse Modeling with TF-IDF Experts Above, we compare ensembling TF-IDF experts in an X-ELM set against choosing a single TF-IDF expert for inference (where a single expert is selected based on the amount of in-language data seen during training). Here, we consider how *sparsifying* the TF-IDF ensemble to m experts holds up against these other settings (Appendix Table C.1). In the cases of m=2,4, this approach sparsifies the ensemble by dynamically selecting the top m experts based on their current ensemble weights, while $m = 1, 8$ are the single expert and full ensemble settings, respectively. We find that for seen languages, reducing the number of experts active to just m=2 usually gives very similar

performance to the full ensemble (m=8). However, this is not true in the case of *unseen* languages, where the m=8 setting consistently outperforms sparser ensembles.

5.6.3 Unseen Languages and Modeling New Languages with X-ELM

We also examine how well X-ELM performs on held-out languages as well as adapts to new languages. Specifically, we consider both zero-shot evaluation and further training of X-ELM on four languages not included in the original XGLM seed model: Azerbaijani (AZ), Hebrew (HE), Polish (PL), and Swedish (SV).[9]

Unseen Language Evaluation We evaluate the existing dense baseline and ensembled TF-IDF clustered experts from the 21B token compute budget (§5.6.2) to test whether continued pretraining with x-BTM improves performance on unseen languages (**X-ELM Training**). We also compare these results to XGLM. We note these models *never* trained on the target languages.

Table 5.4 presents the unseen target language perplexities in the **XGLM** and **X-ELM Training** columns. We find that the original XGLM model performs poorly on the new languages, particularly those less related to XGLM's highest-resourced ones (i.e., AZ and HE). While these perplexities remain high in the dense model and TF-IDF ensembles, training (on other languages) with x-BTM provides some performance improvements over the seed model.

Adapting X-ELM to new languages We now consider how well Hierarchical Multi-Round training (**HMR**) works for language adaptive pretraining (LAPT, Chau et al., 2020), which incorporates new target languages into the continued pretraining process. Here, we group each *target* language with a higher-resource *donor* language already in our pretraining set; these are assigned with the language similarity metric used for typological clustering. We seed each new language's expert with an expert specialized to that language's donor; the new expert is then trained on the donor/target language pair. For HMR inference, we evaluate perplexity with the expert trained on that target

[9]Data for these languages is also obtained from mC4, with the same preprocessing as other languages in our experiments.

language; we also evaluate the donor languages to see what benefit, if any, they receive from the adaptation process.

We compare HMR against jointly continuing training on all four new languages and their respective donors in a single model (**Dense**). Each setting builds on models from the 10.5B compute budget: we continue training on the dense baseline for dense LAPT and branch from the donor languages' k=8 typological experts for HMR training.

All of the LAPT settings provide considerable improvements on the new target languages over the unseen language experiments (Table 5.4, **LAPT** columns). The HMR setting outperforms continued dense training on every new language. Furthermore, HMR training removes the risk of *catastrophic forgetting* (Yogatama et al., 2019) in other LAPT schemes, as this process adds new experts to X-ELM rather than changing existing ones.[10]

Lang	XGLM	X-ELM Training		LAPT	
		Dense	TF-IDF$^*_{ens}$	Dense	HMR
Target					
AZ	1467.45	739.58	722.10	65.73	**32.74**
HE	1817.07	685.02	815.96	53.08	**26.21**
PL	211.76	160.70	178.63	17.71	**16.60**
SV	105.27	92.55	99.24	27.37	**26.16**
Donor					
TR	17.81	15.34	14.28	14.69	**12.72**
AR	16.85	15.29	14.56	14.80	**13.52**
RU	13.43	12.52	12.21	12.28	12.02
EN	14.37	12.88	12.73	12.65	12.63

Table 5.4: Perplexity results on unseen target languages and their respective donor languages. Donor language performance is only **bolded** if these results outperform all other X-ELM settings in that language (Table 5.3).

We also find that this setting provides performance gains on two donor languages over the experiments in §5.6.2. This is likely due to further training with more closely related languages for these languages (e.g., performing training on Arabic with Hebrew rather than French), consequently providing a more informative training signal for the higher-resource donor language as well.

[10]This forgetting of known languages occurs in our dense LAPT baseline, with perplexity decreasing by 1.91 points on average for languages not included in the adaptation setting.

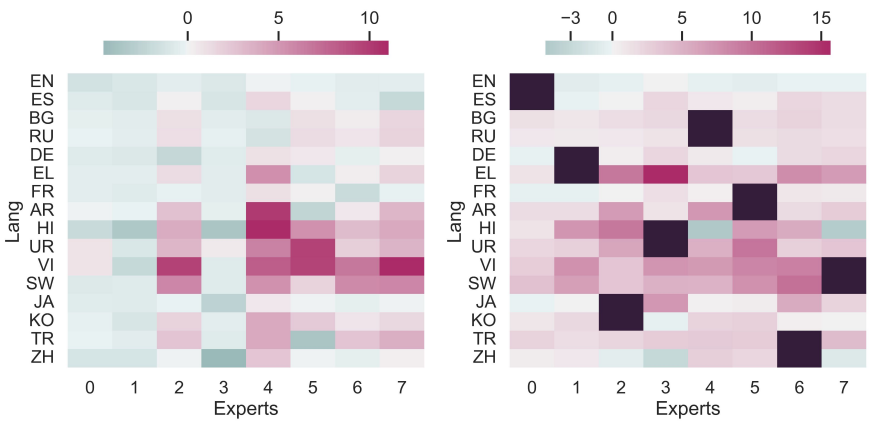


Figure 5.6: Heatmap comparing individual X-ELM perplexities to the seed LM with TF-IDF (left) and Typ. (right) clustering. Positive scores indicate that the expert *forgot* that language. For Typ. clusters, languages that the model was explicitly trained on are grayed out.

5.6.4 X-ELM Forgetting

The preceding sections evaluate X-ELMs as an ensemble of models by dynamically choosing the best expert for a given evaluation setting or ensembling the experts' outputs. However, each expert is initialized with a model trained on all the languages we consider. This prompts the question: how much do individual experts *forget*[11] about the languages they are not specialized to?

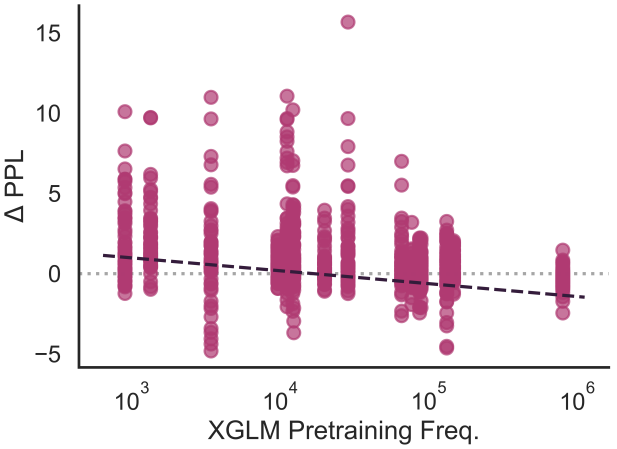


Figure 5.7: Per-expert deltas compared to the original XGLM-1.7B of every pretraining language plotted against the language's frequency in the original XGLM pretraining corpus ($\rho = -0.33, p << 0.001$).

Forgetting occurs as X-ELMs become more specialized. We compare the perplexity of each expert model on all pretraining languages to that of the seed model, XGLM-1.7B (Figure 5.6 for $k = 8$ expert setting). Across the considered values of k, we see less forgetting in the X-ELMs trained on TF-IDF clusters than in those clustered typologically. For the $k = 8$ expert setting, the

[11] We consider an expert to have *forgetten* information about a language if its perplexity on that language increases.

TF-IDF experts only forget on 47.7% of settings, and when forgetting occurs, the perplexity increase over the baseline is 3.10 on average. For typologically clustered experts, these measures are 83.6% and 3.14, respectively; we observe similar trends for the $k = 4$ and $k = 16$ X-ELMs. This implies that though in some cases only small quantities of data are shared across TF-IDF clusters, these data mitigate forgetting over the hard cluster assignments made by typological clustering.

X-ELMs are more likely to forget certain languages. For example, English is rarely forgotten, with only 25% of experts performing worse than the baseline. In comparison, 94.6% of experts perform worse on Urdu than XGLM. One potential cause of this discrepancy is the frequency with which the language was seen during seed training: languages that are more common in the XGLM pretraining corpus see fewer cases of forgetting and have smaller perplexity increases when it does occur (Figure 5.7). Another likely factor is inaccurate language classification in the BTM training data, which is a common issue when training language models on specific languages (Chapter 3); this could lead to related, higher-resourced languages contaminating the datasets for lower-resourced ones (Kreutzer et al., 2022).

5.7 In-Context Learning Experiments

We also measure whether the perplexity improvements from X-ELMs correspond to better performance on downstream tasks. We test the performance of our X-ELMs on three tasks through an in-context learning (ICL) framework, showing that the X-ELM language modeling gains do translate to ICL improvements over the baseline models.

5.7.1 Experimental Setup

We test the in-context learning abilities of X-ELM on three downstream tasks:

XNLI (Conneau et al., 2018) is a multilingual natural language inference benchmark covering 14 of our 16 pretraining languages (excluding JA and KO). Since there are no gold training examples

for XNLI, we use the test set for evaulation and sample demonstrations from the validation set.

XStoryCloze (Lin et al., 2022) is a manually translated benchmark extending StoryCloze (Mostafazadeh et al., 2016) to other languages. This is a story-completion task wherein the model identifies the correct final sentence of a short story. This dataset covers seven of our pretraining languages and four other low-resource languages.

PAWS-X (Yang et al., 2019) is a binary classification task that requires the model to determine whether a pair of sentences are paraphrases. This benchmark covers seven of our pretraining languages, including two (JA and KO) not covered by the other ICL benchmarks.

We reimplement the evaluation protocol from Lin et al. (2022), where the model scores multiple versions of every example (with the different possible labels filled in), and the label of the highest-scoring version is considered as the model's prediction. We use the English prompt formats and evaluation protocols developed for the seed LM of our experts, XGLM, for the downstream tasks of XNLI, XStoryCloze, and PAWS-X. The prompt templates we use are reproduced in Table 5.5.

We compare the performance of X-ELM against dense baselines in both zero- and few-shot learning settings. For all benchmarks, we evaluate on 1,000 random examples, and for few-shot evaluations, we perform five evaluation runs with different demonstration samples and report the average performance. Unless otherwise stated, we evaluate performance on the development set and sample demonstrations from the training set. All few-shot experiments are performed with eight random demonstrations. As we are testing the cross-lingual abilities of X-ELM, these demonstrations are in English for every target language.

Dataset	Prompt	Labels
XNLI	`{Sentence 1}, right? [Mask], {Sentence 2}`	Entailment: `Yes` | Neural: `Also` | Contradiction: `No`
XStoryCloze	`{Context} [Mask]`	Identity
PAWS-X	`{Sentence 1}, right? [Mask], {Sentence 2}`	True: `Yes` | False: `No`

Table 5.5: Prompts used for the ICL experiments in §5.7; the `[MASK]` is filled with one of the label forms given in the last column. For XStoryCloze, `{Context}` refers to the format `{Sent. 1} {Sent. 2} {Sent. 3} {Sent. 4}`, and "Identity" refers to the text of one of the answers given for that example.

	Model	XNLI		XStoryCloze		PAWS-X	
		Acc.	Win Rate	Acc.	Win Rate	Acc.	Win Rate
Zero-shot	XGLM (1.7B)	44.88	28.6%	57.76	**28.6%**	48.54	14.3%
	Dense	44.31	7.1%	56.10	0.0%	48.44	28.6%
	Typ. (TRG)	44.17	7.1%	57.79	**28.6%**	49.86	**42.9%**
	TF-IDF (Top-1)	43.77	14.3%	**57.80**	**28.6%**	**50.04**	28.6%
	TF-IDF (Ens.)	**45.10**	**42.9%**	57.46	14.3%	49.93	0.0%
Few-shot	XGLM (1.7B)	42.34	28.6%	53.21	0.0%	54.52	0.0%
	Dense	41.70	0.0%	55.00	0.0%	54.81	14.3%
	Typ. (TRG)	42.15	†14.3%	54.62	†**71.4%**	55.39	†28.6%
	Typ. (EN)	42.43	†7.1%	**55.54**	†28.6%	55.13	14.3%
	TF-IDF (Top-1)	42.55	21.4%	55.03	†14.3%	**55.50**	†**42.9%**
	TF-IDF (Ens.)	**42.93**	**35.7%**	54.72	28.6%	54.57	14.3%

Table 5.6: Average performance and the percentage of languages where this setting outperforms the others (Win Rate) on the overlap of task evaluation languages and the X-ELM target languages. The **few-shot** setting provides k=8 English demonstrations to the model and averages performance across five runs. †indicates (best) performance ties between two evaluation settings on a language.

5.7.2 Results

We evaluate our best X-ELM setting by perplexity—k=8 experts trained on the larger compute budget of 21B training tokens—on the downstream tasks. Table 5.6 summarizes the results of these evaluations on the languages covered by the X-ELM models. The X-ELM models outperform both the seed model and the compute-matched dense baseline across the three tasks and in both the zero- and few-shot evaluation settings.

Furthermore, though X-ELM improves over the seed model, the dense model underperforms XGLM. This may be due to using different data from the original XGLM pretraining; data quality issues have been previously documented for mC4 (Kreutzer et al., 2022; Chung et al., 2023). We also note that XNLI and XStoryCloze few-shot performance is consistently lower than in the zero-shot setting; this is a recurring issue in multilingual ICL also observed in the base model (Lin et al., 2022).

5.8 Discussion

This work presents an approach to mitigate the *curse of multilinguality* by extending sparse language modeling to the multilingual setting with X-ELMs (cross-lingual expert language models). We find that X-ELMs achieve better perplexity over standard, dense language models trained with the same compute budget; these experts can also be efficiently adapted to new languages without the risk of catastrophic forgetting. X-ELMs also present other benefits over dense models for multilingual modeling, such as not disproportionally benefitting high-resource languages over lower-resourced ones. Finally, we show that these language modeling improvements transfer to downstream tasks.

While our experiments show that X-ELM outperforms dense LMs, we foresee many avenues of future work to further tailor sparse modeling to multilinguality. These include better methods for data allocation—such as clustering methods that leverage cross-lingual signal— and algorithmic improvements to better allocate compute and more effectively ensemble models at inference. By proving the efficacy of sparse language modeling in the multilingual setting, we hope to inspire future work in this vein that fairly models every language while leveraging the potential of cross-lingual learning.

Chapter 6

Conclusion

While multilingual language models have significantly improved NLP applications outside of English, these models often require poorly understood tradeoffs for individual language performance. The work presented in this dissertation presents probing methods for interpreting the features learned by NLP models and analyzes specific aspects of these multilingual models – namely, their data usage and training processes – to develop a better understanding of current multilingual models and their limitations. We then use insights from these analyses to propose new modeling methods mitigating the limitations of current multilingual LMs. This chapter concludes this dissertation with a discussion of the implications of these findings (§6.1) and the proposal of future work that builds on these findings to improve and equalize multilingual NLP (§6.2).

6.1 Discussion

Multilingual LMs learn from tiny subsets of data Many results in this dissertation demonstrate that language models are extremely good at learning from small subsets of their training data, even when trained at massive scales. Chapter 3 finds this to be true in the case of supposedly English-only language models, which learn to model other languages from tiny amounts of data leaked into English text corpora, and Chapter 4 shows that multilingual LMs fit quickly to the linguistics of *all* languages the model is trained on, including in the case of the lowest-resourced ones. These

findings provide evidence that upsampling low-resource languages is not required for the model to learn them.[1] This helps to explain the current success of "English-centric" language models, trained on minimally filtered webcrawled data without consideration of the text's language, over some more carefully curated multilingual large language models.

Perhaps more surprisingly, these models also learn cross-lingual transfer and specific NLP tasks from these small amounts of incidental supervision. Our manual analysis in Chapter 3 found examples of bitext in the contaminated data, which we hypothesized may provide cross-lingual supervision from the language modeling objective. Briakou et al. (2023) then supported this hypothesis by controlling for the amount of these examples included in LM training and testing its effect on transfer. Similarly, in Blevins et al. (2023), we found that pretraining data for other models contains examples of labeled task data, including task data in many different languages. Given these models' aptitude for learning from even tiny pieces of their training data, it remains crucial to document and build methods for interpreting these large text corpora.

Multilingual pretraining is stochastic and complex Chapter 4 shows that while there are general trends for knowledge acquisition in multilingual learning, many of the specifics for individual languages are less consistent. For example, while the overall order in which linguistic skills are learned mirrors the pattern found in prior work on English pretraining dynamics (Liu et al., 2021), this ordering does not often hold for learning cross-lingual skills for any given language pair. Our statistical analysis confirms this, showing that many of the training dynamics we consider are difficult to predict from factors such as language resourcefulness and similarity. While there may be unconsidered factors causing the observed behavior, the difficulty in predicting pretraining patterns suggests that (1) these patterns are likely influenced by the inherent randomness of pretraining and (2) we will likely observe different patterns of behavior on different training runs, particularly for lower resource languages that are more affected by the randomness of data sampling.

Another unexpected phenomenon observed in multilingual pretraining dynamics is that of *model forgetting*, where the model's performance on a given language or task decreases during the

[1] though this upsampling is often still beneficial for these languages (Downey et al., 2024)

pretraining process. The observed layer-wise shift of knowledge explains some of this phenomenon; however, for many languages, the final checkpoint of the model still performs worse than prior versions regardless of the considered layer. We hypothesize that this phenomenon — along with the general instability of learning described above — is due to the previously described *curse of multilinguality* (Conneau et al., 2020a). Specifically, these behaviors seem to be evidence of this curse developing throughout the pretraining process, leading to the observed behavior in the final model state.

Cross-lingual expert language models benefit all languages The hypothesis motivating X-ELMS (Chapter 5) is that the curse of multilinguality can be mitigated by explicitly allocating model capacity (through different expert LMs) to related subsets of the multilingual training set. The experiments prove this to be overwhelmingly true: all considered languages greatly benefit from this setup. Notably, grouping languages by linguistic similarity outperforms automatic clustering methods. This observation mirrors similar findings on targeted language modeling with dense models (Ogueji et al., 2021; Ogunremi et al., 2023; Downey et al., 2024, inter alia), as well as the results in (Chang et al., 2023), which finds that *related* cases of multilingual training benefit lower-resource languages.

X-ELM also brings other benefits beyond performance to multilingual language modeling. Adapting X-ELM sets to new languages by training a new expert is much easier than adapting a dense multilingual model, which risks forgetting previously learned information during continued training (Yogatama et al., 2019). Furthermore, due to the modularity and independent training of the experts, multilingual branch-train-merge provides efficiency benefits and flexibility during training over dense LM training; we also see similar flexibility when using X-ELM for inference. These qualities mean that X-ELMs are more accessible to train, adapt, and use in low-compute settings, which we hope will help democratize multilingual language modeling and increase access to these technologies in currently underserved languages.

It is unclear how well the current iteration of X-ELM performs cross-lingual transfer. The

primary evidence in this setting comes from the downstream evaluations, which show that clustering data by language performs worse than automatic clustering (despite observing the opposite trend on intrinsic LM evaluations). This is likely due to the soft TF-IDF allowing for some cross-lingual text sharing during BTM training. Therefore, a primary focus of future work on X-ELM will be to improve the cross-lingual alignment of the experts, as discussed in the next section.

6.2 Future Work

Breaking the Curse of Multilinguality The most significant limitation in current multilingual LMs is the *curse of multilinguality*. X-ELMs address this *curse* at the model parameter stage, but other factors can also limit the overall effectiveness of multilingual LMs. For instance, the choice of tokenization can lead to over-segmentation of low-resource languages and scripts (e.g., Ahia et al., 2023). While recent work has proposed new methods for *adapting* embeddings to new languages during model specialization (Dobler and De Melo, 2023; Downey et al., 2023), the underlying tension of tokenizing many different languages with a limited vocabulary remains. One promising direction in this space is byte-level tokenization for multilingual modeling (Xue et al., 2022), which removes the need for data-driven tokenization methods. However, UTF-8 byte encodings still show bias towards higher-resource Latin script languages; Limisiewicz et al. (2024) address this by combining byte-level representation and data-driven morphological compression to represent all languages within the model's scope fairly. Incorporating fairer input representations into X-ELMs through methods such as these will be vital in building a fully equitable multilingual system.

There are also potential improvements at the individual model level, whether working with a dense language model or X-ELMs. For instance, Downey et al. (2024) recently analyzed the factors that improve cross-lingual learning when training LMs on related languages, similar to how data is allocated to experts during multilingual BTM. Using the best practices recommended here would likely lead to increased improvements for the considered lower-resourced languages. Finally, one of the current drawbacks of X-ELM is a reduced ability to transfer across languages.

As this is an essential benefit of multilingual language modeling, an important next step is to address this limitation, such as by developing better data allocation methods containing more cross-lingual information or by introducing additional training objectives to keep the experts aligned for cross-lingual ensembling.

Better Data Protocols for Languages Beyond English A significant benefit of multilingual LMs is their ability to transfer information cross-lingually, and this skill mitigates *some* of the data gap between English and other languages. However, data limitations for most of the world's languages remain an important issue: even with the perfect cross-lingual transfer, the current focus on English data (and translations of English data) limits what the model learns and thus underrepresents information relevant to other language speakers.[2] Furthermore, current multilingual data quantity and quality issues extend to pretraining corpora (Kreutzer et al., 2022), limiting the ability of these models to learn in-language and cross-lingual representations.

Therefore, improving data collection and annotation continues to be an important area of multilingual NLP. This applies to all aspects of multilingual data collection, ranging from pretraining data to the annotation of supervised training data and task evaluation data. Some potential improvements in current approaches for these areas include (1) widening the net for gathering multilingual data, such as considering multimodal data (in particular, speech) in low-resource languages where text data is scarce, and (2) interacting with the language communities these multilingual technologies will serve to narrow down the types of models (and therefore, the types of data) that will be most beneficial to their needs. However, these approaches do not remove the underlying need for increased efforts to gather high-quality, manually annotated data in other languages — such as the ongoing work by Masakhane NLP[3] and the Universal NER project (Mayhew et al., 2023) — rather than the noisy conversion of data to other languages from English.

[2]For example, there are many Wikipedia articles with no corresponding English article, such as those on Jean-Joseph Sanfourche (a French artist) and Torsten Haß (a German author), as of May 2024.

[3]https://www.masakhane.io/

Bibliography

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140. Association for Computational Linguistics.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872. Association for Computational Linguistics.

Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. Interpretability and analysis in

neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Online. Association for Computational Linguistics.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10. Asian Federation of Natural Language Processing.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono- and cross-lingual pretraining dynamics of multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663.

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19.

Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A. Smith, and Luke Zettlemoyer. 2024. Breaking the curse of multilinguality with cross-lingual expert language models. *arXiv preprint arXiv:2401.10440*.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Eleftheria Briakou, Colin Cherry, and George Foster. 2023. Searching for needles in a haystack: On

the role of incidental bilingualism in palm's translation capability. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. 2023. When is multilinguality a curse? language modeling for 250 high-and low-resource languages. *arXiv preprint arXiv:2311.09205*.

Ethan C Chau, Lucy H Lin, and Noah A Smith. 2020. Parsing with multilingual bert, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Ethan A Chi, John Hewitt, and Christopher D Manning. 2020a. Finding universal grammatical relations in multilingual bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577.

Zewen Chi, Li Dong, Furu Wei, Xianling Mao, and He-Yan Huang. 2020b. Can monolingual pretrained models help cross-lingual classification? In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 12–17.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-Yi Lee. 2020. Pretrained language model

embryology: The birth of albert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828.

David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226. Association for Computational Linguistics.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The grammar-learning trajectories of neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8281–8297, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

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

Jacob Delvin. 2019. Multilingual BERT Readme. `https://github.com/google-research/bert/blob/master/multilingual.md`.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2021. When is bert multilingual? isolating crucial ingredients for cross-lingual transfer. *arXiv preprint arXiv:2110.14782*.

Tim Dettmers and Luke Zettlemoyer. 2019. Sparse networks from scratch: Faster training without losing performance. *CoRR*, abs/1907.04840.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Konstantin Dobler and Gerard De Melo. 2023. Focus: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cm Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 268–281.

CM Downey, Terra Blevins, Dhwani Serai, Dwija Parikh, and Shane Steinert-Threlkeld. 2024. Targeted multilingual adaptation for low-resource language families. *arXiv preprint arXiv:2405.12413*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR)*.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for bert's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 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)*, pages 4555–4567.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. 2020. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 434–452.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Evangelia Gogoulou, Ariel Ekgren, Tim Isbister, and Magnus Sahlgren. 2021. Cross-lingual transfer of monolingual models. *arXiv preprint arXiv:2109.07348*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It's not greek to mbert: Inducing word-level translations from multilingual bert. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 45–56.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL-HLT*.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual language model dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2440–2452, Marseille, France. European Language Resources Association.

Suchin Gururangan, Dallas Card, Sarah K Drier, Emily K Gade, Leroy Z Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A Smith. 2022a. Whose language counts as high quality? measuring language ideologies in text data selection. *arXiv preprint arXiv:2201.10474*.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022b. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Suchin Gururangan, Margaret Li, Mike Lewis, Weijia Shi, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2023. Scaling expert language models with unsupervised domain discovery.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *International Conference on Machine Learning*.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference of Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *The Tenth Machine Translation Summit Proceedings of Conference*, pages 79–86. International Association for Machine Translation.

Dan Kondratyuk and Milan Straka. 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)*, pages 2779–2795.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of*

the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4483–4499.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models.

Zuchao Li, Kevin Parnow, Hai Zhao, Zhuosheng Zhang, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2021. Cross-lingual transferring of pre-trained contextualized language models. *arXiv preprint arXiv:2107.12627*.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. Myte: Morphology-driven byte encoding for better and fairer multilingual language modeling. *arXiv preprint arXiv:2403.10691*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative

language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association of Computational Linguistics*, 4:521–535.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A Smith. 2021. Probing across time: What does roberta know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, et al. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *arXiv preprint arXiv:2305.13169*.

Shixiang Lu, Wei Wei, Xiaoyin Fu, and Bo Xu. 2012. Translation model based cross-lingual language model adaptation: from word models to phrase models. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 512–522.

Koena Ronny Mabokela, Madimetja Jonas D Manamela, and Mabu Manaileng. 2014. Modeling code-switching speech on under-resourced languages for language identification. In *SLTU*, pages 225–230.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.

Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915.

David Mareček. 2008. *Automatic Alignment of Tectogrammatical Trees from Czech-English Parallel Corpus*. Charles University, MFF UK.

Antonio Martínez-García, Toni Badia, and Jeremy Barnes. 2021. Evaluating morphological typology in zero-shot cross-lingual transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3136–3153, Online. Association for Computational Linguistics.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Šuppa, Hila Gonen, Joseph Marvin Imperial, Börje F Karlsson, Peiqin Lin, Nikola Ljubešić, LJ Miranda, et al. 2023. Universal ner: A gold-standard multilingual named entity recognition benchmark. *arXiv e-prints*, pages arXiv–2311.

Hesham Mostafa and Xin Wang. 2019. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4646–4655. PMLR.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual bert. In *EACL 2021-The 16th Conference of the European Chapter of the Association for Computational Linguistics*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126.

Tolulope Ogunremi, Dan Jurafsky, and Christopher D Manning. 2023. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1221–1236.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Isabel Papadimitriou, Ethan A Chi, Richard Futrell, and Kyle Mahowald. 2021. Deep subjecthood: Higher-order grammatical features in multilingual bert. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532.

Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Laura Pérez-Mayos, Miguel Ballesteros, and Leo Wanner. 2021. How much pretraining data do language models need to learn syntax? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1571–1582.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings*

of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3479–3495.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer.

Daniel Pimienta, Daniel Prado, and Álvaro Blanco. 2009. Twelve years of measuring linguistic diversity in the internet: Balance and perspectives. *United Nations Educational, Scientific and Cultural Organization.*

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. The necessity of syntactic parsing for semantic role labeling. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 5, pages 1117–1123.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Machel Reid and Mikel Artetxe. 2022. PARADISE: Exploiting parallel data for multilingual

sequence-to-sequence pretraining. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. Bert is not an interlingua and the bias of tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 47–55.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. 2023. Transfer to a low-resource language via close relatives: The case study on faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737.

Leandro Rodrigues de Souza, Rodrigo Nogueira, and Roberto Lotufo. 2021. On the ability of monolingual models to learn language-agnostic representations. *arXiv preprint arXiv:2109.01942*.

Ekaterina Taktasheva, Vladislav Mikhailov, and Ekaterina Artemova. 2021. Shaking syntactic trees on the sesame street: Multilingual probing with controllable perturbations. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 191–210.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information and Communication Technology Research*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ke Tran. 2020. From english to foreign languages: Transferring pre-trained language models. *arXiv preprint arXiv:2002.07306*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Yulia Tsvetkov, Sunayana Sitaram, Manaal Faruqui, Guillaume Lample, Patrick Littell, David R Mortensen, Alan W Black, Lori Levin, and Chris Dyer. 2016. Polyglot neural language models: A case study in cross-lingual phonetic representation learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1357–1366.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *CoRR*.

Ulla Uebler. 2001. Multilingual speech recognition in seven languages. *Speech communication*, 35(1-2):53–69.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about bert's layers? a closer look at the nlp pipeline in monolingual and multilingual models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350.

Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.

Genta Indra Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Ping Xu and Pascale Fung. 2012. Cross-lingual language modeling with syntactic reordering for low-resource speech recognition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 766–776.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer.

In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 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)*, pages 3687–3692.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Appendix A

Additional Materials for Chapter Three

A.1 Full Results of the Automatic Language Identity Analysis

We present a more complete set of results for the automatic language composition analysis (Section 3.2) in Table A.1. We include every language that has 10,000 or more tokens in at least one of the considered corpora; we additionally report numbers for Basque and Frisian, as both languages are included in the experiments in Section 3.3.

A.2 Full Results of Transfer Experiments

Full results for whole word MLM are given in Table A.3; results for POS probing can be found in Table A.4 and results for finetuned POS tagging are detailed in Table A.5.

ISO	Language	Number of Tokens							
		Wiki	Book	Stories	OpenWebText	CCNews	C4	BERT	RoBERTa
en	English	2.0B	802.4M	6.2B	6.4B	13.0B	17.8B	2.8B	28.3B
sq	Albanian	3.3K	0	195	8.8K	42.5M	14.4K	3.3K	42.5M
es	Spanish	112.8K	120.4K	150.6K	3.4M	36.6M	5.9M	233.2K	40.3M
de	German	176.2K	5.4K	104.8K	3.1M	34.4M	9.0M	181.5K	37.8M
ro	Romanian	19.6K	174	6.4K	1.4M	28.7M	164.0K	19.8K	30.2M
pt	Portugese	43.2K	760	44.2K	1.5M	10.0M	1.9M	44.0K	11.5M
it	Italian	102.9K	3.9K	46.1K	1.6M	9.2M	2.6M	106.7K	10.9M
fr	French	201.1K	88.1K	126.6K	2.5M	7.2M	6.0M	289.2K	10.1M
pl	Polish	56.2K	51	5.0K	239.9K	5.3M	686.9K	56.2K	5.6M
nl	Dutch	28.0K	1.0K	37.3K	254.7K	4.4M	1.7M	29.0K	4.8M
vi	Vietnamese	25.5K	98	2.8K	10.5K	3.5M	277.6K	25.6K	3.6M
tl	Tagalog	3.2K	3.7K	28.7K	124.3K	3.1M	312.1K	6.9K	3.3M
cs	Czech	8.8K	12	2.1K	152.7K	2.0M	295.0K	8.8K	2.1M
fi	Finnish	6.9K	119	4.7K	243.2K	1.7M	214.5K	7.0K	1.9M
no	Norwegian	9.5K	170	6.4K	204.3K	1.6M	300.5K	9.7K	1.8M
hu	Hungarian	8.9K	51	5.6K	32.5K	1.6M	194.2K	9.0K	1.7M
hi	Hindi	6.7K	0	520	32.2K	1.5M	328.0K	6.7K	1.6M
hr	Croatian	4.0K	0	482	313.2K	1.2M	30.8K	4.0K	1.5M
id	Indonesian	1.5K	100	12.9K	83.5K	1.3M	997.7K	1.6K	1.4M
ru	Russian	17.4K	606	3.9K	956.3K	64.8K	2.3M	18.0K	1.0M
sv	Swedish	11.1K	567	9.3K	784.9K	236.3K	743.5K	11.6K	1.0M
sr	Serbian	753	0	709	39.0K	976.2K	36.8K	753	1.0M
et	Estonian	2.8K	0	288	8.0K	817.1K	32.0K	2.8K	828.2K
tr	Turkish	6.3K	541	9.4K	131.4K	535.0K	401.9K	6.9K	682.6K
af	Afrikaans	852	0	2.7K	6.7K	584.1K	145.3K	852	594.3K
ku	Kurdish	185	0	0	6.7K	468.0K	3.5K	185	474.9K
da	Danish	3.1K	20	5.3K	249.8K	157.8K	271.1K	3.1K	415.9K
gl	Galican	101	0	309	637	317.5K	9.6K	101	318.6K
ja	Japanese	5.8K	3.4K	23.8K	188.7K	76.3K	3.0M	9.2K	298.1K
ca	Catalan	5.2K	99	418	28.2K	258.8K	108.3K	5.3K	292.8K
ar	Arabic	5.3K	0	665	154.2K	89.6K	601.7K	5.3K	249.7K
ko	Korean	3.2K	20	45	208.1K	8.0K	4.1M	3.3K	219.4K
el	Greek	15.2K	777	1.8K	123.8K	28.4K	288.7K	16.0K	169.9K
sl	Slovenian	262	0	250	102.1K	14.5K	46.8K	262	117.1K
is	Icelandic	1.5K	65.8K	758	10.4K	11.1K	114.7K	67.2K	89.5K
ga	Irish	1.2K	0	839	8.7K	77.9K	468.4K	1.2K	88.6K
uk	Ukranian	3.5K	10	232	63.4K	3.5K	232.1K	3.5K	70.7K
he	Hebrew	5.2K	0	4.6K	46.5K	9.6K	138.0K	5.2K	66.0K
lt	Lithuanian	3.4K	12	1.1K	2.8K	54.9K	56.8K	3.4K	62.2K
sk	Slovak	1.9K	0	76	16.2K	43.9K	64.7K	1.9K	62.0K
ms	Malay	896	29	1.4K	1.8K	45.9K	42.8K	925	50.0K
sw	Swahili	44	16	533	143	47.7K	5.9K	60	48.5K
eo	Esperanto	461	114	2.6K	34.9K	7.2K	37.2K	575	45.2K
zh	Chinese	4.1K	12	5.5K	30.8K	4.5K	410.2K	4.1K	44.8K
lv	Latvian	1.4K	0	367	4.0K	38.5K	47.0K	1.4K	44.3K
bn	Bengali	2.5K	24.8K	51	6.2K	10.1K	48.6K	27.3K	43.6K
fa	Persian	3.0K	0	261	28.2K	5.8K	668.9K	3.0K	37.2K
nn	Norysk	283	0	0	4.5K	32.5K	4.6K	283	37.2K
la	Latin	6.0K	641	3.8K	19.4K	4.7K	34.7K	6.6K	34.5K
az	Azerbaijani	2.0K	0	55	884	27.1K	12.9K	2.0K	30.0K
th	Thai	7.6K	0	592	15.1K	5.5K	131.8K	7.6K	28.7K
bg	Bulgarian	6.7K	20	284	18.7K	2.8K	96.7K	6.7K	28.5K
cy	Welsh	1.2K	84	440	18.5K	7.4K	59.5K	1.3K	27.6K
ilo	Iloko	19	0	16	628	18.8K	1.1K	19	19.5K
ur	Urdu	5.0K	0	24	6.4K	7.0K	27.9K	5.0K	18.4K
ta	Tamil	5.4K	0	234	5.9K	5.8K	42.2K	5.4K	17.4K
mt	Maltese	177	0	0	371	13.9K	20.3K	177	14.5K
hy	Armenian	2.6K	0	0	7.5K	2.9K	21.5K	2.6K	13.0K
gd	Gaelic	198	0	52	874	8.8K	117.6K	198	10.0K
eu	Basque	99	5	1.8K	2.8K	2.4K	19.3K	104	7.0K
fy	Frisian	80	0	1.3K	1.5K	601	9.8K	80	3.4K
Total (Non-En)		983k	322k	682k	18.6M	201M	406M†	1.3M	222M

Table A.1: Full results for the automatic language composition analysis of pretraining corpora presented in Section 3.2. The last two columns include the total data sizes for BERT and RoBERTa; T5 was trained on C4; † represents the projected estimate for the full dataset.

ISO	Monolingual			Multilingual	
	BERT	RoBERTa	T5	mBERT	XLMR
ar	2.91 (0.60%)	3.11 (0.0%)	1.91 (**41.26%**)	1.95 (0.05%)	1.73 (9.8e-4%)
bg	3.03 (0.06%)	3.25 (0.0%)	2.88 (**17.83%**)	1.93 (0.67%)	1.72 (1.2e-3%)
ca	1.95 (0.01%)	1.83 (0.0%)	2.08 (1.23%)	1.55 (0.12%)	1.56 (4.6e-4%)
cs	2.64 (0.03%)	2.64 (0.0%)	2.85 (**10.30%**)	2.00 (0.22%)	1.86 (5.6e-4%)
da	2.19 (0.01%)	2.07 (0.0%)	2.42 (3.56%)	1.74 (0.14%)	1.63 (4.5e-4%)
de	2.21 (0.02%)	2.14 (0.0%)	1.85 (0.26%)	1.65 (0.37%)	1.67 (1.5e-3%)
el	2.74 (0.36%)	2.96 (0.0%)	1.82 (**40.30%**)	2.05 (0.05%)	1.73 (1.4e-3%)
en	1.38 (0.03%)	1.32 (0.0%)	1.44 (0.15%)	1.37 (0.22%)	1.42 (1.3e-3%)
es	1.76 (0.01%)	1.67 (0.0%)	1.88 (1.48%)	1.37 (0.11%)	1.37 (8.2e-4%)
et	3.02 (0.03%)	2.92 (0.0%)	3.35 (1.55%)	2.47 (0.33%)	2.21 (1.5e-3%)
fa	2.80 (1.12%)	3.34 (0.0%)	2.00 (**42.14%**)	1.70 (0.05%)	1.55 (6.1e-3%)
fi	3.18 (8.3e-3%)	3.06 (0.0%)	3.48 (0.13%)	2.45 (0.35%)	2.24 (9.9e-4%)
fr	1.90 (0.01%)	1.81 (0.0%)	1.78 (0.26%)	1.53 (0.43%)	1.57 (8.5e-4%)
he	2.82 (0.72%)	3.08 (0.0%)	1.97 (**40.30%**)	2.05 (0.06%)	1.89 (4.2e-4%)
hi	1.98 (**12.06%**)	2.84 (0.0%)	1.64 (**42.82%**)	1.64 (0.05%)	1.39 (1.3e-3%)
hr	2.38 (7.1e-3%)	2.27 (0.0%)	2.57 (3.71%)	1.85 (0.08%)	1.73 (1.1e-3%)
hu	2.78 (0.02%)	2.72 (0.0%)	3.00 (2.60%)	2.12 (0.28%)	1.93 (1.1e-3%)
id	2.34 (0.05%)	2.22 (0.0%)	2.54 (0.13%)	1.70 (0.12%)	1.59 (4.5e-3%)
it	1.92 (0.01%)	1.83 (0.0%)	2.05 (0.42%)	1.51 (0.10%)	1.52 (1.0e-3%)
ja	34.41 (**39.97%**)	47.67 (0.0%)	9.50 (**22.19%**)	35.22 (0.05%)	31.30 (0.03%)
ko	1.60 (**59.65%**)	4.78 (0.0%)	2.32 (**38.63%**)	2.65 (0.25%)	2.53 (0.03%)
lt	3.06 (0.80%)	3.12 (0.0%)	3.41 (8.78%)	2.48 (0.97%)	2.23 (7.7e-3%)
lv	2.89 (0.48%)	2.84 (0.0%)	3.13 (**12.52%**)	2.36 (0.30%)	2.06 (2.8e-3%)
ms	2.34 (0.03%)	2.21 (0.0%)	2.53 (0.10%)	1.71 (0.10%)	1.58 (1.9e-3%)
nl	2.18 (8.0e-3%)	2.04 (0.0%)	2.31 (0.21%)	1.64 (0.08%)	1.62 (4.6e-4%)
no	2.24 (0.03%)	2.10 (0.0%)	2.49 (3.01%)	1.74 (0.13%)	1.66 (2.0e-3%)
pl	2.60 (9.3e-3%)	2.60 (0.0%)	2.83 (6.50%)	1.96 (0.44%)	1.88 (7.2e-4%)
pt	1.86 (0.02%)	1.76 (0.0%)	2.01 (2.05%)	1.45 (0.11%)	1.43 (1.0-3%)
ro	2.03 (0.01%)	2.02 (0.0%)	1.73 (0.18%)	1.63 (0.25%)	1.54 (7.9e-4%)
ru	3.05 (0.02%)	3.25 (0.0%)	2.90 (**21.1%**)	1.92 (0.53%)	1.82 (2.1e-3%)
sk	2.86 (0.05%)	2.81 (0.0%)	3.14 (7.08%)	2.20 (0.19%)	2.00 (1.4e-3%)
sl	2.37 (9.7e-3%)	2.24 (0.0%)	2.53 (3.45%)	1.91 (0.06%)	1.73 (1.1e-3%)
sr	3.01 (0.71%)	3.33 (0.0%)	2.95 (**17.14%**)	1.95 (0.19%)	1.77 (4.8e-4%)
sv	2.57 (7.9e-3%)	2.40 (0.0%)	2.77 (2.08%)	1.90 (0.15%)	1.80 (8.5e-4%)
th	2.13 (**36.91%**)	11.79 (0.0%)	2.73 (**28.58%**)	8.34 (0.12%)	5.42 (1.6e-3%)
tl	2.14 (0.10%)	2.02 (0.0%)	2.44 (0.18%)	1.81 (0.12%)	1.70 (2.9e-3%)
tr	2.94 (0.01%)	2.87 (0.0%)	3.19 (7.36%)	2.13 (0.31%)	1.91 (2.0e-3%)
uk	3.36 (0.52%)	3.73 (0.0%)	3.23 (**24.12%**)	2.11 (0.54%)	1.94 (1.4e-3%)
vi	1.76 (1.44%)	1.95 (0.0%)	1.89 (**15.12%**)	1.19 (0.08%)	1.16 (3.2e-3%)

Table A.2: The average number of subword tokens per white-spaced word (and the percentage of UNKed out tokens) in the Wiki40b validation set for each language. Cases where more than 10% of tokens are unked out are in bold.

ISO	Monolingual				Multilingual		
	BERT$_{ba}$	**BERT**$_{lg}$	**RoBERTa**$_{ba}$	**RoBERTa**$_{lg}$	**mBERT**	**XLMR**$_{ba}$	**XLMR**$_{lg}$
ar	6.214	9.331	3.319	3.899	1.849	1.871	1.691
bg	6.334	7.883	3.544	3.587	1.553	1.494	1.358
ca	3.382	3.565	1.834	1.640	1.108	1.477	1.329
cs	4.316	4.738	2.634	2.493	1.703	1.715	1.533
da	3.560	3.832	2.104	1.931	1.420	1.427	1.272
de	3.430	3.644	1.815	1.634	1.102	1.361	1.218
el	6.934	8.915	3.852	3.885	1.793	1.588	1.440
en	1.285	1.377	0.595	0.516	0.938	1.249	1.131
es	3.281	3.551	1.526	1.345	1.036	1.284	1.165
et	3.846	4.108	2.448	2.318	1.878	1.858	1.671
fa	5.813	8.501	3.614	4.113	1.723	1.567	1.418
fi	3.732	4.064	2.357	2.240	1.633	1.618	1.451
fr	3.213	3.439	1.586	1.414	1.038	1.434	1.305
he	6.490	9.074	3.530	3.831	1.817	1.976	1.739
hi	4.240*	5.503*	1.487	1.407	1.876	1.641	1.516
hr	3.972	4.298	2.267	2.109	1.563	1.644	1.484
hu	4.203	4.585	2.741	2.632	1.778	1.713	1.548
id	3.436	3.665	1.976	1.838	1.221	1.243	1.129
it	3.263	3.536	1.661	1.475	1.098	1.402	1.256
ja	1.840*	2.065*	5.481	6.775	2.082	6.827	8.016
ko	0.781*	0.846*	4.204	4.639	3.144	3.504	3.241
lt	3.953	4.271	2.746	2.633	1.840	1.789	1.604
lv	4.231	4.512	2.833	2.730	1.890	1.750	1.548
ms	3.461	3.698	2.010	1.886	1.280	1.365	1.257
nl	3.445	3.693	1.855	1.680	1.222	1.397	1.257
no	3.580	3.873	2.052	1.872	1.398	1.469	1.312
pl	4.020	4.505	2.506	2.365	1.495	1.604	1.437
pt	3.442	3.718	1.658	1.465	1.128	1.316	1.190
ro	3.641	3.929	1.950	1.772	1.402	1.435	1.286
ru	6.747	8.122	3.624	3.673	1.385	1.491	1.344
sk	4.263	4.628	2.714	2.594	1.804	1.753	1.594
sl	3.972	4.294	2.415	2.273	1.642	1.563	1.391
sr	6.081	7.216	3.610	3.661	1.772	1.783	1.681
sv	3.774	4.081	2.196	2.019	1.460	1.523	1.372
th	1.551*	1.689*	3.312	3.535	3.861	2.119	2.237
tl	3.250	3.458	1.763	1.623	1.616	1.713	1.572
tr	4.102	4.427	2.715	2.585	1.635	1.603	1.460
uk	6.542	7.912	3.763	3.823	1.566	1.635	1.488
vi	5.134	5.794	2.590	2.574	1.046	1.191	1.055

Table A.3: Full results for the zero-shot BPC experiments in Section 3.3. Results noted with * correspond to cases of high UNK rates in the tokenization of the data (Section 3.4).

ISO	Baselines		Monolingual					Multilingual		
	Maj. Label	Word Maj.	BERT$_{ba}$	BERT$_{lg}$	Ro$_{ba}$	Ro$_{lg}$	T5-base	mBERT	XLMR$_{ba}$	XLMR$_{lg}$
af	21.650	83.335	81.695	84.855	88.858	92.324	90.464	93.590	97.490	96.381
ar	33.297	90.148	79.595	79.994	78.939	79.242	43.182	93.724	95.659	95.533
bg	21.834	86.091	85.617	84.238	78.514	81.147	85.304	94.977	97.240	97.103
ca	17.868	90.984	93.208	93.432	94.333	94.545	95.863	97.610	98.222	98.202
cs	24.708	91.284	82.312	80.591	89.272	93.488	86.560	96.554	97.548	97.744
cy	31.099	73.587	69.625	72.641	69.171	70.309	76.220	81.713	76.690	77.215
da	18.606	77.841	81.137	81.485	85.308	91.057	86.832	91.901	96.757	96.087
de	17.784	81.992	86.663	88.306	91.266	93.074	92.878	92.022	94.851	94.020
el	21.148	81.128	79.996	79.110	66.604	76.795	37.050	92.509	95.435	95.371
en	16.999	82.920	93.803	92.903	94.785	94.418	96.286	93.666	95.366	95.342
es	17.734	90.734	89.639	92.860	97.652	93.171	97.222	97.699	98.418	98.497
et	26.462	78.486	74.987	78.524	74.460	81.150	79.287	91.891	90.717	91.143
eu	24.422	77.591	73.152	75.663	72.713	72.254	80.294	84.712	88.744	88.023
fa	33.521	91.916	78.545	78.202	67.668	66.767	46.485	93.025	96.310	96.597
fi	27.965	74.378	71.083	72.301	77.318	82.587	77.630	92.621	95.823	95.872
fr	18.749	89.584	90.960	90.197	93.690	95.278	96.612	95.963	95.837	95.813
fy	14.815	85.190	79.749	82.128	78.766	78.216	86.351	90.898	87.908	89.405
ga	29.122	81.512	72.470	77.009	76.983	79.169	82.430	87.097	91.493	92.583
gd	21.166	80.114	78.264	79.641	74.989	76.281	79.590	78.387	84.102	85.609
gl	22.969	86.294	87.727	89.058	92.638	93.176	93.647	92.559	95.145	95.548
he	23.601	85.491	75.040	75.393	69.930	70.160	45.758	93.206	96.403	94.864
hi	22.128	89.365	68.650	68.861	77.250	80.597	38.185	94.054	95.524	94.250
hr	24.182	83.533	80.408	82.370	92.955	94.742	87.784	96.164	98.011	98.313
hu	22.429	60.356	72.619	73.444	76.403	83.633	79.868	88.273	93.404	91.868
hy	24.995	68.931	50.956	52.033	58.560	58.792	44.142	89.001	90.403	93.937
id	21.642	81.278	4.151	4.151	79.539	81.664	4.151	4.151	4.151	4.151
is	17.286	90.407	80.969	84.217	79.792	82.154	84.121	91.414	97.459	97.778
it	19.920	89.758	89.497	91.317	93.991	95.521	94.715	96.662	97.454	96.854
ja	30.137	85.592	76.268	75.659	83.491	85.013	39.409	92.362	90.844	91.080
ko	30.011	67.715	48.090	47.683	67.852	70.745	47.288	76.359	80.372	80.704
la	21.355	94.888	91.416	94.079	92.133	92.489	95.928	95.008	98.250	97.548
lt	31.345	61.009	67.026	70.382	67.892	66.602	77.164	90.542	91.641	94.710
lv	27.108	79.198	71.889	77.190	72.061	74.857	81.630	87.627	93.565	92.775
mt	19.489	76.131	75.582	78.313	75.199	75.156	80.854	76.877	70.113	74.191
nl	16.799	81.878	79.448	84.540	90.126	92.816	89.207	94.356	95.920	95.990
pl	24.900	83.868	82.357	81.721	91.563	92.491	90.494	94.184	98.231	97.840
pt	18.117	83.517	85.594	86.801	91.591	92.114	95.120	94.066	96.981	94.810
ro	24.849	85.537	82.630	84.414	93.830	95.332	93.407	95.284	97.443	97.229
ru	23.843	88.593	84.258	85.526	82.713	86.811	88.812	95.299	96.943	94.714
sk	19.264	61.821	80.441	83.025	86.850	89.339	86.872	92.414	96.290	95.810
sl	21.289	77.815	82.459	80.959	88.357	88.176	87.189	96.388	97.953	98.144
sr	24.378	82.523	84.930	85.434	94.762	94.769	89.884	92.727	98.638	98.333
sv	17.579	78.993	71.818	79.061	78.523	88.662	83.226	92.826	96.246	95.350
ta	29.389	53.042	43.992	40.563	38.361	43.228	43.147	74.912	76.521	75.606
tr	36.494	82.343	78.115	73.015	67.656	67.340	80.830	88.633	91.392	89.939
uk	23.213	71.964	78.214	78.950	65.442	69.780	80.780	91.836	96.213	96.823
ur	23.564	85.736	68.112	68.819	69.202	67.458	34.116	88.954	91.170	92.341
vi	32.019	75.901	54.764	54.267	54.054	56.402	60.768	75.844	85.855	81.554
zh	27.478	78.696	49.462	51.400	64.537	67.238	44.479	87.363	88.565	85.921

Table A.4: Full results for the frozen POS tagging experiments in Section 3.3.

ISO	Monolingual		Multilingual	
	BERT_{ba}	Ro_{ba}	mBERT	XLMR_{ba}
af	92.108	94.528	96.802	97.158
ar	92.896	93.417	96.151	96.673
bg	97.185	96.797	98.714	99.145
ca	98.058	98.264	98.813	98.845
cs	98.191	98.306	98.803	98.952
cy	82.152	72.484	92.109	84.927
da	93.134	93.872	97.037	97.556
de	93.285	93.554	95.178	95.189
el	92.725	90.452	96.735	96.897
en	96.496	97.186	96.431	97.082
es	97.841	98.459	98.828	98.781
et	95.142	95.244	96.547	97.402
eu	91.313	90.117	94.529	94.956
fa	94.477	94.113	97.193	97.609
fi	93.534	93.436	96.275	97.823
fr	96.970	97.112	97.845	98.102
fy	92.383	92.770	95.557	95.721
ga	91.263	91.164	93.293	94.334
gd	91.774	90.220	92.814	93.797
gl	94.261	95.565	95.130	96.892
he	91.373	90.847	96.242	97.035
hi	83.566	94.437	96.554	97.384
hr	95.955	96.687	98.061	98.293
hu	83.886	85.540	94.763	94.035
hy	53.953	86.965	92.985	93.829
id	4.151	91.635	4.151	4.151
is	96.583	96.489	97.899	98.404
it	96.521	97.297	98.172	98.334
ja	86.477	93.707	96.600	97.112
ko	47.783	91.514	94.941	95.464
la	98.386	98.633	99.399	99.199
lt	82.707	84.507	93.026	94.636
lv	93.252	93.710	95.744	96.908
mt	87.284	85.603	89.248	86.349
nl	93.974	94.861	96.669	96.969
pl	96.604	97.064	98.569	98.980
pt	95.420	96.572	97.526	97.611
ro	95.795	96.044	97.568	97.878
ru	97.310	97.103	98.306	98.568
sk	93.608	93.994	97.282	97.373
sl	95.233	95.776	98.157	98.798
sr	96.140	97.154	98.531	98.802
sv	90.737	92.805	96.242	97.080
ta	42.363	49.452	77.185	64.354
tr	92.390	92.320	94.667	94.946
uk	93.320	93.980	96.020	96.975
ur	84.556	84.432	92.622	93.251
vi	46.954	50.874	89.653	91.261
zh	55.811	84.877	95.022	95.972

Table A.5: Full results for the finetuned POS tagging experiments in Section 3.3.

Appendix B

Additional Materials for Chapter Four

B.1 Expanded Layer-wise Analysis

This section expands on the layer-wise analysis of XLM-R$_{replica}$ presented in §4.5. Figure B.1 gives additional layer-wise heatmaps over time. Figure B.2 shows the expected layer (i.e., average layer weighted by relative performance) of XLM-R$_{replica}$ at different time steps. The expected layer decreases over time: by 1.79, 1.61, 1.08, and 2.79 for CS, EN, HI, and JA respectively on dependency arc classification; and by 2.49, 2.25, 0.43, and 0.77 for BG, EN, HI, and ZH respectively on XNLI.

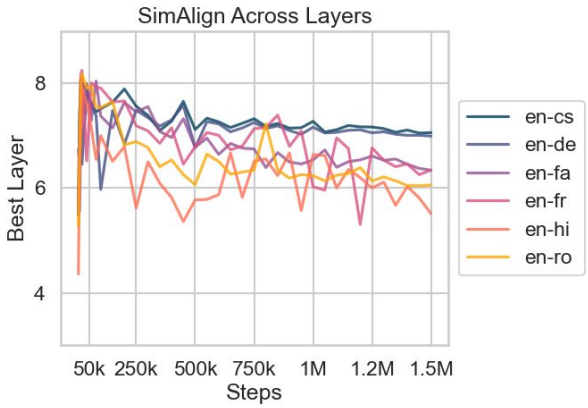


Figure B.4: Change in the expected best layer for word alignment via SimAlign over time in XLM-R$_{replica}$

We also provide additional examples of layer-wise cross-lingual transfer in Figure B.3; we find that for cross-lingual transfer, the best internal layer outperforms the best final layer state on average by 7.67 on arc classification transfer, 3.39 on XNLI, and 14.2 F1 on Simalign. Figure B.4 shows the change in the expected best layer over time for SimAlign.

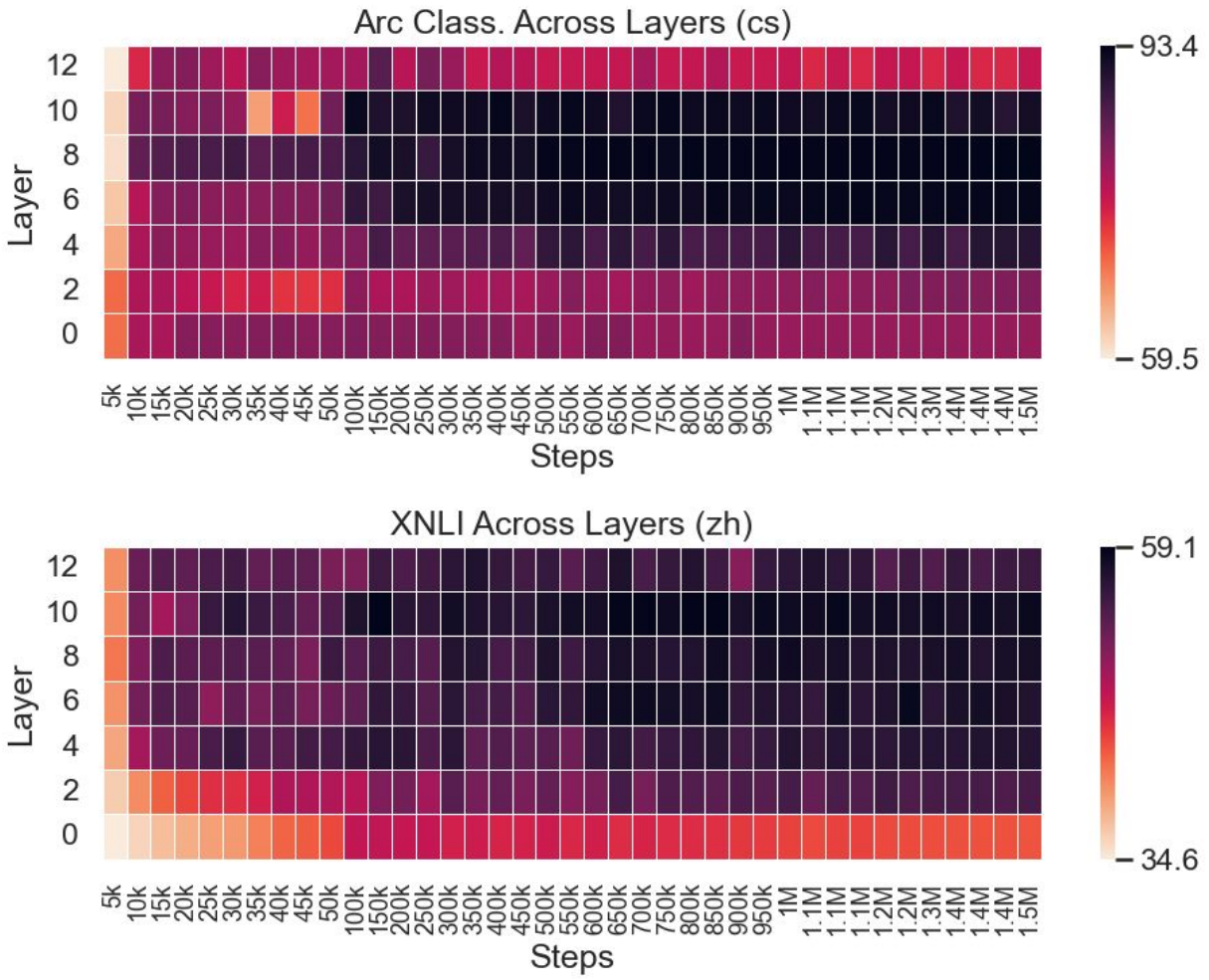


Figure B.1: Layer-wise performance heatmaps for Czech arc classification and Chinese XNLI.

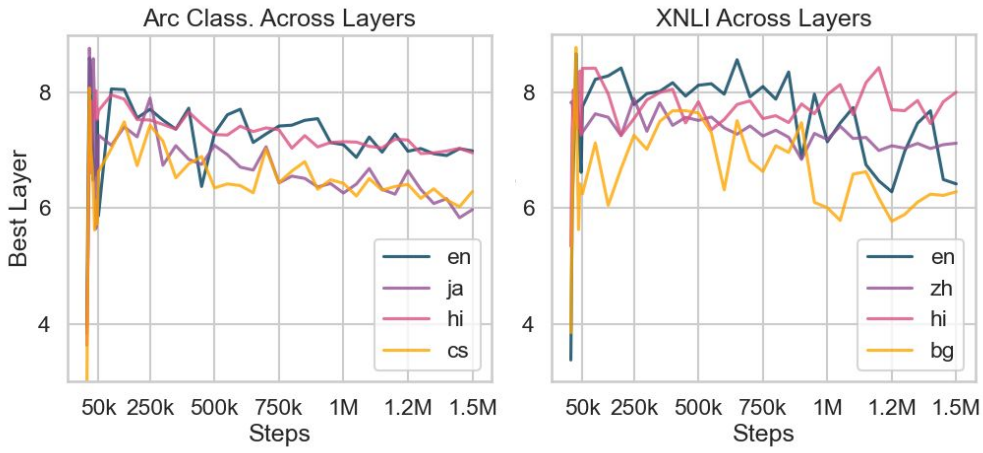


Figure B.2: The expected best layer for in-language dependency arc classification and XNLI over time on XLM-R$_{replica}$.

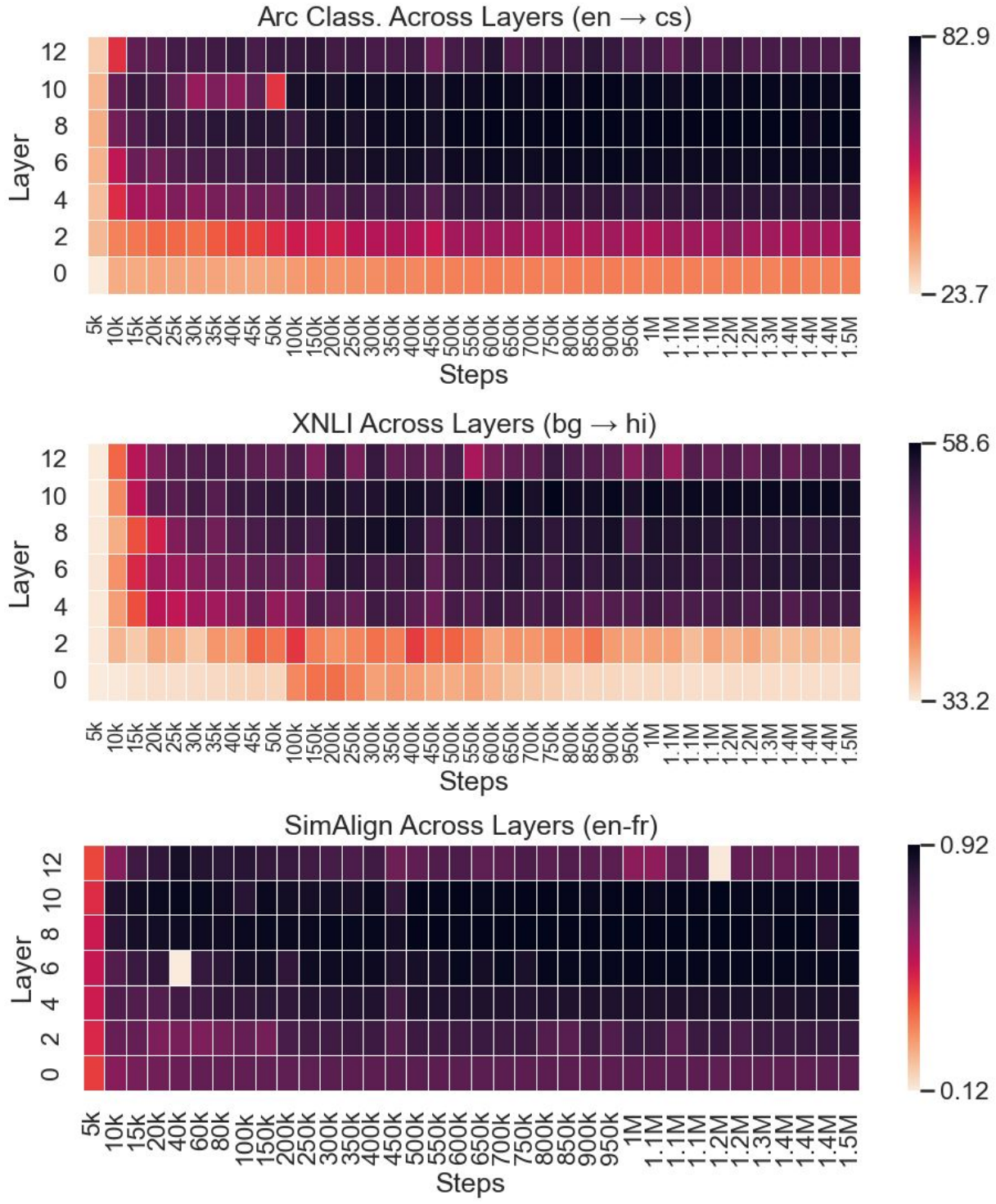


Figure B.3: Additional heatmaps of cross-lingual transfer at different layers and timesteps of XLM-R$_{replica}$.

B.2 Additional Across Time Analyses

This section includes additional results from our analysis of knowledge acquisition during multilingual pretraining:

- Figure B.5 presents BPC learning curves for each language in the CC100 training data.
- Figure B.7 covers the learning progress of XLM-R$_{replica}$ on dependency arc prediction, arc classification, and XNLI, expanding on the results in §4.3.2.
- Figure B.6 gives the relative performance for in-language POS and XNLI across training checkpoints discussed in §4.3.2.
- Figure B.8 presents more detailed results for relative performance over time when transferring out of English. This expands on the summary figures discussed in §4.4.2.

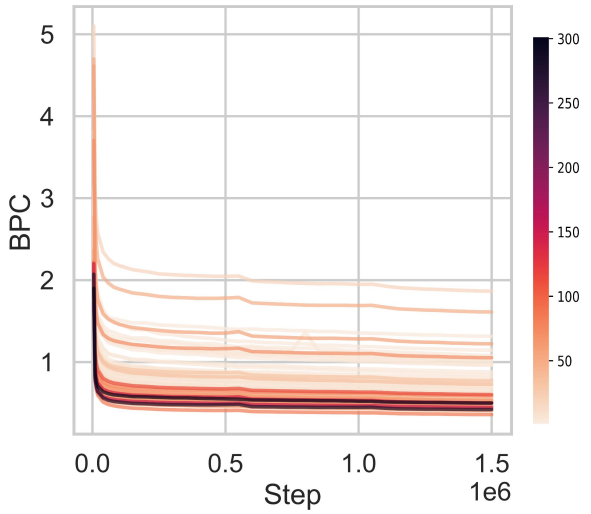


Figure B.5: Learning Curves for BPC in each training language. Lines are colored by the amount of pretraining data available for that language.

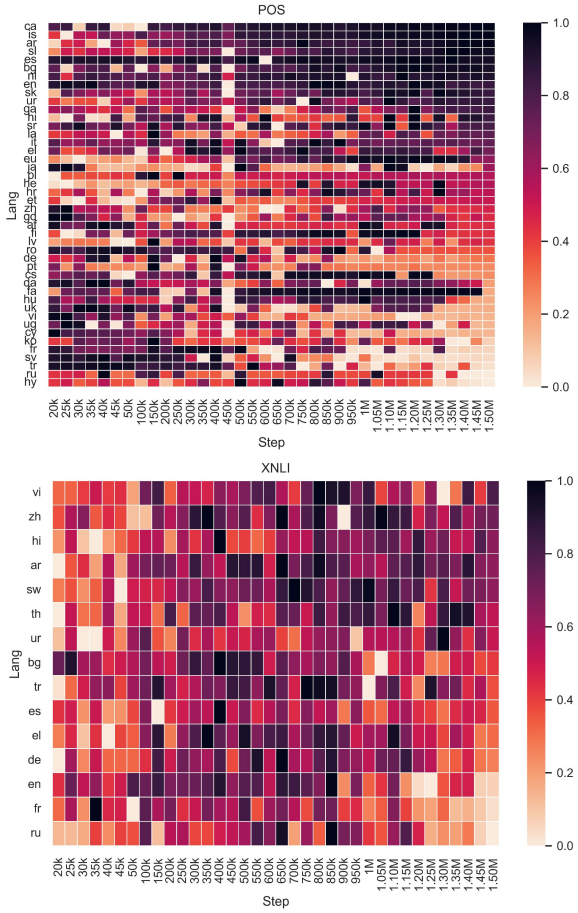


Figure B.6: Heatmap of relative performance over time for different languages for POS tagging and XNLI. Languages are ordered by the amount of performance degradation at the final checkpoint.

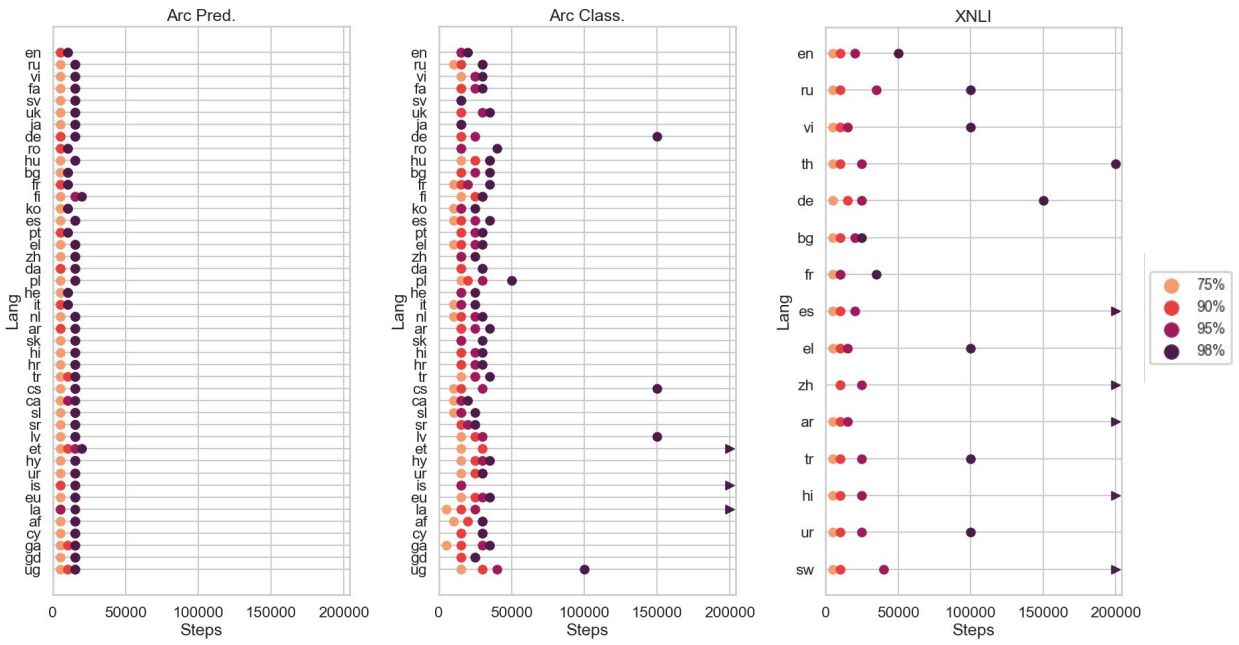


Figure B.7: Learning Progress of XLM-R$_{replica}$ across training, up to 200k training steps. Each point represents the step at which the model achieves x% of the best overall performance of the model on that task.

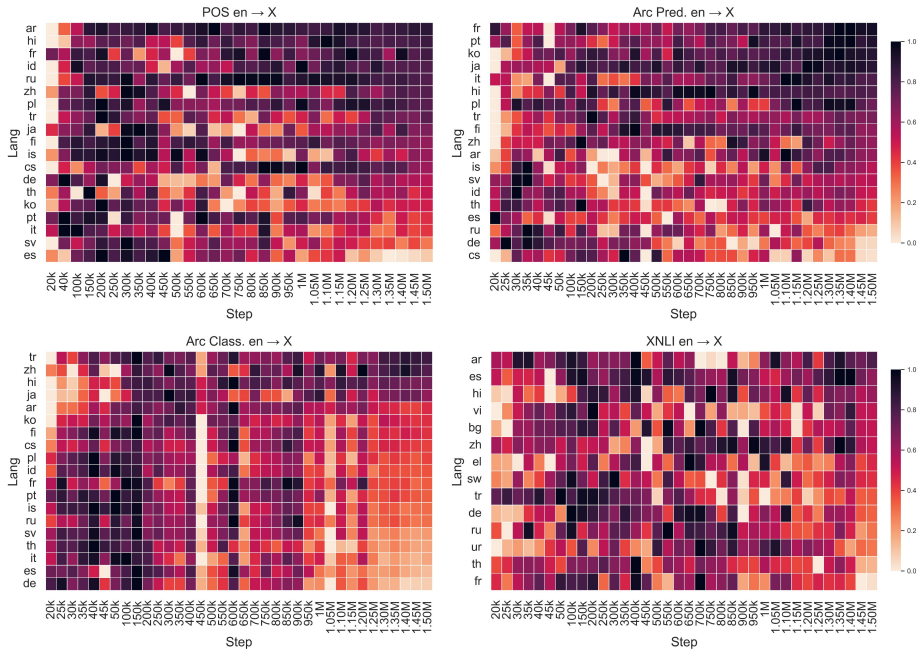


Figure B.8: Heatmap of relative performance over time for cross-lingual transfer with English as the source language. Languages are ordered by the amount of performance degradation at the final checkpoint.

Appendix C

Additional Materials for Chapter Five

C.1 Additional X-ELM Analysis

C.1.1 Hierachical Multi-Round (HMR) Training for Seen Languages

The final column of Table C.2 evaluates the effectiveness of HMR training for continued training of X-ELMs. Here, we select the $k = 4$ typologically clustered expert from the 10.5B token compute setting that covers the pair of languages as the seed model for each $k = 8$ setting. We then adapt on the $k = 8$ **Typ.** setting for another 10.5B tokens. Our results find that this adaptation scheme *underperforms* the **Typ.** experts trained from the seed model for 21B tokens, indicating that it is more effective to train X-ELMs in a single run rather than dividing the compute budget across two rounds of training. We hypothesize that this negative result is due to the XGLM seed model, which is a fully pretrained model, already learning adequate transfer across language families; we leave further investigation of this to future work. However, this finding is in contrast with §5.6.3, which shows that the HMR scheme is very effective for adapting experts to *unseen* languages.

C.2 Full Experimental Results

Table C.2 presents the full perplexity results for the $k = 4$ and $k = 16$ X-ELM experiments, trained on a 10.5B token compute budget. We find that both choices of k underperform the $k = 8$ setting.

Figure C.1 reports the per-expert forgetting relative to the baseline model (XGLM-1.7B) in the $k = 4$ and $k = 16$ settings. On average, the $k = 4$ TF-IDF experts experience forgetting in only 18.8% of cases with an average perplexity increase of 1.24 when forgetting occurs; the typology experts forget 78.1% of the time with an average perplexity increase of 1.34. For the $k = 16$ setting, these statistics are 60.9% and 0.9 for the TF-IDF clusters and 89.4% and 1.24 for the typology clusters.

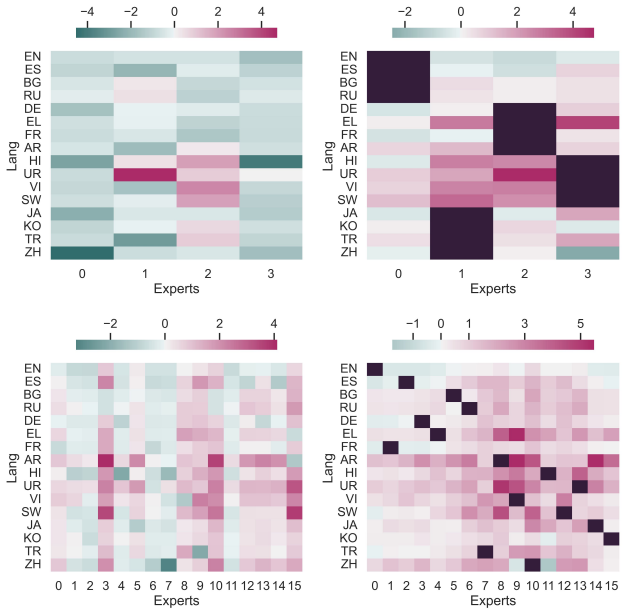


Figure C.1: Heatmap of X-ELM forgetting with TF-IDF (left) and Typ. (right) clustering, from the $k = 4$ (top) and $k = 16$ (bottom) settings.

Downstream Evaluation on Individual Languages Tables C.3, C.4, and C.5 detail the per-language results for XNLI, XStoryCloze, and PAWS-X, respectively.

Lang.	TF-IDF Ens.			
	top-1	m=2	m=4	m=8
AR	14.00	14.12	14.05	14.05
BG	10.27	10.27	10.27	10.27
DE	12.95	13.09	13.07	13.04
EL	9.03	9.03	8.99	9.00
EN	12.68	12.50	12.48	12.47
ES	13.54	13.40	13.39	13.37
FR	10.79	10.92	10.88	10.88
HI	14.36	13.47	13.62	13.62
JA	11.36	11.35	11.37	11.37
KO	7.61	7.53	7.53	7.53
RU	11.83	11.90	11.90	11.90
SW	19.04	18.67	18.67	18.67
TR	13.41	13.58	13.58	13.58
UR	13.26	13.52	13.52	13.52
VI	10.56	10.41	10.41	10.42
ZH	12.61	12.84	12.84	12.87
Avg.	12.33	12.29	12.29	12.28
AZ	–	736.49	724.97	722.10
HE	–	749.12	719.68	719.05
PL	–	177.31	175.27	174.83
SV	–	95.33	94.37	94.14

Table C.1: Perplexity scores of the different inference methods on the TF-IDF X-ELMs trained with 21B tokens. **Top-1** chooses a single expert per language, with no routing mechanism, whereas **m=2,4,8** ensembles TF-IDF experts.

Lang.			k=4 Experts			k=16 Experts		
	XGLM	Dense	TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.	TF-IDF$_{top1}$	TF-IDF$_{ens}$*	Typ.
AR	16.85	15.29	14.99	15.03	15.00	15.60	15.67	15.40
BG	11.31	10.44	10.39	10.39	10.42	11.10	10.70	10.31
DE	15.53	14.02	13.85	13.89	13.71	14.71	14.43	14.5
EL	10.44	9.40	9.36	9.33	9.28	9.72	9.64	9.41
EN	14.37	12.88	12.64	12.71	12.78	13.60	13.23	13.27
ES	16.02	14.13	13.93	13.96	14.06	14.83	14.58	14.59
FR	13.12	11.78	11.62	11.65	11.55	12.38	12.13	12.15
HI	18.28	14.28	14.22	14.21	12.64	16.11	15.67	13.86
JA	14.57	12.31	12.23	12.12	11.73	13.39	13.14	13.18
KO	8.82	7.78	7.81	7.77	7.70	8.14	8.09	7.75
RU	13.43	12.52	12.30	12.33	12.46	12.96	12.76	12.82
SW	19.85	18.70	18.61	18.62	18.19	19.38	19.13	16.43
TR	17.81	15.34	14.85	14.96	14.81	15.67	15.78	15.52
UR	14.38	13.45	13.56	13.73	13.18	13.88	13.87	12.65
VI	13.07	11.39	11.43	11.21	10.32	11.85	11.65	11.59
ZH	17.91	13.74	13.38	13.70	13.11	14.65	14.95	13.58
Avg.	14.74	12.97	12.82	12.85	12.56	13.62	13.46	12.94

Table C.2: Per-language and average perplexity results for the $k = 4$ and $k = 16$ X-ELM experiments (original XGLM and $k = 1$ dense model included for comparison). Lower numbers are better. Each X-ELM setting is trained on 10.5B tokens. *TF-IDF ensemble uses more parameters for inference than other evaluations.

Model	AR	BG	DE	EL	EN	ES	FR	HI	RU	SW	TH*	TR	UR	VI	ZH
Zero-shot															
XGLM (1.7B)	46.8	45.7	44.1	42.5	51.5	36.5	47.2	45.9	47.3	43.6	44.9	42.5	43.5	43.9	46.9
Dense	47.9	45.0	45.3	45.2	51.1	37.2	45.9	44.5	44.5	39.6	44.3	44.8	43.1	41.6	44.6
Typ. (TRG)	46.2	44.9	43.9	45.4	52.0	36.0	47.2	43.5	41.9	40.6	–	44.2	41.9	44.4	46.3
TF-IDF (Top-1)	47.3	45.1	42.9	47.1	51.5	36.3	45.6	43.1	40.6	38.7	–	45.0	43.2	41.8	44.6
TF-IDF (Ens.)	48.6	47.2	46.2	43.1	53.0	37.0	47.5	45.7	45.6	40.0	45.8	44.1	44.2	42.6	46.6
Few-shot															
XGLM (1.7B)	42.0	44.2	43.4	43.4	47.2	38.1	45.5	40.4	43.1	41.4	41.9	38.0	39.7	42.2	44.3
Dense	43.4	42.2	43.6	41.9	45.9	36.7	42.3	42.2	40.8	40.0	43.2	39.9	40.3	41.0	43.5
Typ. (TRG)	42.8	43.0	42.6	43.0	47.3	38.5	45.4	38.9	39.9	41.7	–	41.0	39.6	42.9	43.4
Typ. (EN)	42.2	42.6	44.0	42.6	47.3	38.5	42.9	42.1	42.8	40.9	44.5	41.1	40.0	42.1	44.9
TF-IDF (Top-1)	43.1	43.6	43.2	41.7	47.5	38.2	45.3	42.1	40.5	41.9	–	41.1	41.4	42.1	44.1
TF-IDF (Ens.)	43.0	43.3	44.3	43.3	47.8	37.7	44.2	43.2	42.3	41.4	44.4	41.8	41.0	42.7	44.9

Table C.3: Individual language accuracy on XNLI. *TH (Thai) is an unseen language for the X-ELM models.

Model	AR	EN	ES	EU*	HI	ID*	MY*	RU	SW	TE*	ZH
Zero-shot											
XGLM (1.7B)	53.3	63.1	57.3	56.4	55.0	59.3	54.0	60.0	60.1	57.0	55.5
Dense	50.5	60.7	56.1	52.1	52.0	55.4	53.4	58.6	58.6	55.5	56.2
Typ. (TRG)	52.3	62.7	57.5	–	52.7	–	–	60.2	60.3	–	58.8
TF-IDF (Top-1)	52.1	62.1	58.1	53.2	55.2	57.7	52.6	59.6	60.5	57.3	57.0
TF-IDF (Ens.)	51.9	60.4	57.8	54.0	55.4	58.5	52.0	59.5	60.2	57.1	57.0
Few-shot											
XGLM (1.7B)	48.6	58.2	53.2	51.7	50.4	52.1	51.5	52.5	56.0	56.5	53.7
Dense	50.2	59.0	54.6	51.3	51.6	53.5	52.9	56.9	57.8	54.2	55.2
Typ. (TRG)	50.3	60.1	55.0	–	52.0	–	–	57.4	58.0	–	56.0
Typ. (EN)	48.8	60.1	55.0	–	52.2	–	–	53.7	57.4	–	55.2
TF-IDF (Top-1)	49.3	59.5	54.5	51.4	52.4	55.2	52.9	55.4	58.0	56.1	56.1
TF-IDF (Ens.)	49.4	59.0	53.8	51.1	52.5	54.5	52.0	55.1	57.8	55.0	55.4

Table C.4: Individual language accuracy on XStoryCloze (and EN StoryCloze). *Unseen languages for the X-ELM models.

Model	DE	EN	ES	FR	JA	KO	ZH
Zero-shot							
XGLM (1.7B)	44.5	47.9	51.8	45.2	53.8	49.6	47.0
Dense	49.4	47.5	50.7	47.5	48.8	47.2	48.0
Typ. (TRG)	47.9	47.9	53.0	45.5	55.4	53.6	45.7
TF-IDF (Top-1)	47.4	46.9	55.0	45.9	54.9	49.4	50.8
TF-IDF (Ens.)	49.1	47.1	52.1	47.2	53.6	50.0	50.4
Few-shot							
XGLM (1.7B)	56.3	50.5	55.4	55.2	55.6	53.0	55.7
Dense	56.0	54.9	55.8	55.2	54.9	53.8	53.0
Typ. (TRG)	56.5	53.4	55.8	55.1	55.6	55.9	55.4
Typ. (EN)	56.0	53.4	55.8	55.4	55.5	54.7	55.1
TF-IDF (Top-1)	56.6	54.2	55.7	54.9	55.6	55.7	55.7
TF-IDF (Ens.)	53.8	54.9	54.8	53.6	55.3	55.2	54.4

Table C.5: Individual language accuracy on PAWS-X.