Adapting Pre-Trained Models and Leveraging Targeted Multilinguality for Under-Resourced and Endangered Language Processing

C.M. Downey

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:

Gina-Anne Levow, Chair

Shane Steinert-Threlkeld, Chair

Fei Xia

Program Authorized to Offer Degree:
Department of Linguistics

University of Washington

Abstract

Adapting Pre-Trained Models
and Leveraging Targeted Multilinguality
for Under-Resourced and Endangered Language Processing

C.M. Downey

Co-Chairs of the Supervisory Committee:
Gina-Anne Levow
Linguistics

Shane Steinert-Threlkeld
Linguistics

Advances in Natural Language Processing (NLP) over the past decade have largely been driven by the scale of data and computation used to train large neural network-based models. However, these techniques are inapplicable to the vast majority of the world's languages, which lack the vast digitized text datasets available for English and a few other very high-resource languages. In this dissertation, we present three case studies for extending NLP applications to under-resourced languages. These case studies include conducting unsupervised morphological segmentation for extremely low-resource languages via multilingual training and transfer, optimizing the vocabulary of a pre-trained cross-lingual model for specific target language(s), and specializing a pre-trained model for a low-resource language family (Uralic). Based on these case studies, we argue for three broad, guiding principles in extending NLP applications to under-resourced languages. First: where possible, robustly pre-trained models and representations should be leveraged. Second: components of pre-trained models that are not optimized for new languages should be substituted or substantially adapted. Third: targeted multilingual training provides a middle ground between the lack of adequate data to train models for individual under-resourced languages on one hand, and the diminishing returns of "massively multilingual" training on the other.

TABLE OF CONTENTS

DEDICATION

To all those without whose love and support I would not have been able to reach this moment: to my family, who have always supported me in following my dreams; to my friends through the years, especially those who shared the PhD journey with me; to my amazing advisors, who invested so much time and effort in guiding me to my goals; and of course to my cat Stevie, whose company kept me going these past few years.

Chapter 1

INTRODUCTION

1.1 Background

The current dominant approach for building Natural Language Processing (NLP) systems is to train large neural networks on very high-resource languages like English, Chinese, or German, for which vast amounts of textual data are available (i.e. hundreds of gigabytes or more: Brown et al., 2020; Touvron et al., 2023, i.a.). These techniques are inapplicable to the majority of the world's languages, which lack the large requisite text datasets. Joshi et al. (2020b) quantify language-wise data availability (Figure 1.1) and introduce a classification system ranging from 0 to 5. Even languages spoken by many millions of speakers such as Urdu, Indonesian, and Tamil fall into level 3 or below, and are typically considered "low/under-resourced" in NLP research. This methodological gap undermines the potentially vital role machine learning can play in creating critical NLP services such as machine translation, automatic speech recognition, and assisted writing. In the face of large-scale language endangerment and loss in the 21st Century (Brenzinger et al., 2003), development of these tools can aid the effort to ensure that a diversity of world languages can thrive in the digital era.

This gulf in data availability obviously cannot be overcome by techniques that rely on data scale. It is also generally accepted that large-scale models cannot be robustly trained without large-scale data (Hoffmann et al., 2022). However, several machine learning-based techniques may offer hope for bridging the data divide and extending NLP to a wider range of the world's languages. Specifically, *unsupervised/self-supervised learning* enables training with smaller amounts of specialized data than supervised paradigms require; *multilingual modeling* allows language data to be pooled by training on more than one language at once; and *transfer learning* leverages existing models trained in higher-resource languages for use with new, low-resource languages.

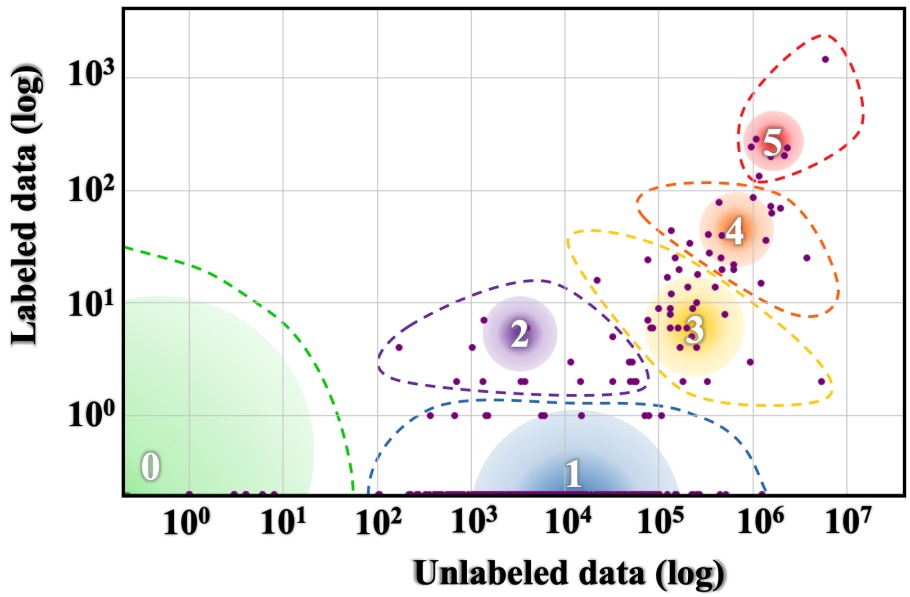


Figure 1.1: Language data "taxonomy" from Joshi et al. (2020b), categorizing languages by their availability of labeled and unlabeled data. English is visible as an outlier in the top right-hand corner.

Unsupervised learning is named in opposition to supervised learning, in which a system is trained to predict a label, classification, or decision over some instance of raw data (e.g. a sentence; see Figure 1.2). This training paradigm requires pairings of raw data with annotated labels ("labeled data", Figure 1.1 Y-axis), to train a mapping from the former to the latter. Unsupervised learning, in contrast, learns patterns from raw data alone. Self-supervised learning — a type of unsupervised learning in which a missing portion of raw data is predicted based on the remainder — is ubiquitous in NLP (Figure 1.2). For text, this is typically referred to as "language modeling": either predicting the next word/unit in a sequence (traditional language modeling), or predicting a missing word (masked language modeling: Devlin et al., 2019).

One advantage of self-supervised modeling is that it does not rely on the availability of labeled data, which is much more scarce and hard to curate than unlabeled (raw) data. However, it is not straightforward to design systems that accomplish a traditionally

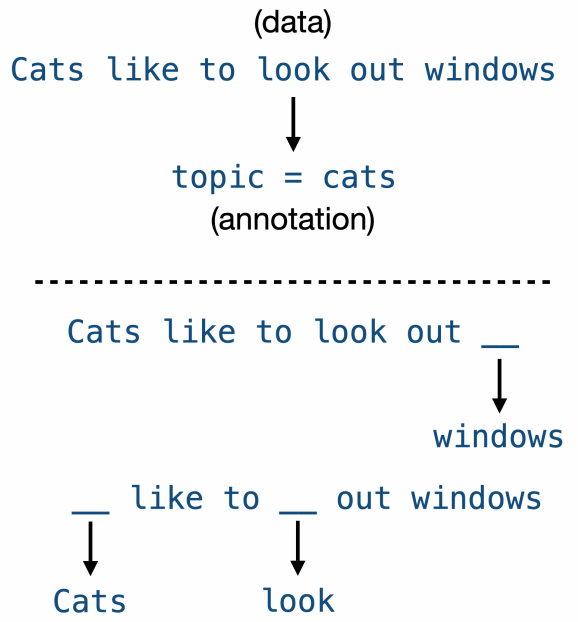


Figure 1.2: Top: an example *supervised* learning task, mapping raw text data to a label (topic). Bottom: examples of *self-supervised* learning, in which missing parts of the raw data are predicted.

supervised task with purely self-supervised training. Instead, self-supervised modeling is usually employed in NLP as a foundation on which to build more robust systems that require less supervised training data (Peters et al., 2018, i.a.). Yann LeCun famously described the relationship between these paradigms through the metaphor of a cake, in which self-supervised learning forms the body of the cake, while supervised learning is the icing.[1] This refers both to the fact that unlabeled data is much more plentiful, and that it provides a richer signal for learning useful data representations (LeCun, 2016).

In this dissertation, self-supervised learning will be used both to leverage unlabeled data for under-resourced languages by training general-purpose language models (Chapters 3 and 4), as well as to accomplish morphological segmentation (Chapter 2). However, the

[1]Reinforcement learning is beyond the scope of this work, but LeCun describes it as the cherry on top of the cake. It has become an important component in successful language model-based chatbots such as ChatGPT and GPT-4 (OpenAI et al., 2024).

models still rely on large amounts of unlabeled data, which most languages lack. Thus, self-supervised learning on its own is not enough to bridge the data disparity with under-resourced languages.

Multilingual modeling is a relatively recent approach in NLP, in which a single model (usually an self-supervised language model) is trained on a set of languages, rather than a single one (Ha et al., 2016; Conneau and Lample, 2019, i.a.). The advantage of this technique is that data can be *pooled* across languages, allowing a model to be trained on much more data in total than would be possible for any of the languages individually. This capability seems to be enabled by the flexibility of vector representations in modern neural network-based language models, which can learn language-general patterns, as well as language-specific ones (Conneau et al., 2020b). These properties have led large multilingual models to become the foundation for most modern NLP applications in languages outside of English (Devlin, 2019; Conneau and Lample, 2019; Conneau et al., 2020a; Liu et al., 2020; Scao et al., 2023; Üstün et al., 2024, i.a.).

An unresolved question in multilingual modeling revolves around the so-called "curse of multilinguality" — the observation that overall model performance decreases as more languages are added during training (Conneau et al., 2020a; Wang et al., 2020b). However, it is generally accepted that under-resourced languages benefit from *some* multilinguality during training, especially when added languages are similar in some way (Conneau et al., 2020a; Ogunremi et al., 2023; Chang et al., 2023). Multilingual modeling is employed in each chapter of this dissertation, though we in general argue for an approach of *targeted* multilinguality, rather than the "massively multilingual" approach common in modern NLP.[2]

Finally, as applied in the multilingual setting, transfer learning is the ability to either 1) directly apply a model trained on a high-resource language to a new (usually low-resource) language, or 2) transfer the model via some limited adaptive training, or by replacing model components with ones specialized for the new language. Transferring a pre-trained model directly to a new language (with no adaptation) is usually referred to as "zero-shot" transfer,

[2]Because the main chapters of this dissertation are multi-authored works, we adopt the first-person plural pronoun *we* throughout. The content of the Introduction and Conclusion are authored by the dissertation author alone (C.M. Downey).

and is an incredibly powerful tool for extending NLP applications to low-resource languages (Conneau et al., 2020a, i.a.). However, a more realistic scenario is one in which there is a small amount of data available for a low-resource language of interest, in which "few-shot" transfer (Lin et al., 2022) or "language-adaptive training" (Chau et al., 2020) can be used to shift the model towards better performance in the new language.

Despite the success of cross-lingual transfer learning, it is still widely debated which languages facilitate the best transfer to which others. Common explanations for transfer success include typological or genealogical relatedness between "source" and "target" languages (Lin et al., 2019; Chang et al., 2023), overlap in the script (orthography) used for each (Muller et al., 2021), multilinguality of the training set (Conneau et al., 2020a), or language-general patterns that make even monolingual models converge on similar representations (Artetxe et al., 2020).

Transfer learning, including zero-shot transfer, is a core component of each study in this dissertation. In particular, we will examine the ways in which multilingual training interacts with typological and genealogical language relatedness to facilitate the better transfer to low-resource languages.

1.2 Outline of Chapters

Each chapter in this dissertation presents an experimental study of methodologies for improving NLP performance in under-resourced languages. As highlighted in the previous section, self-supervised learning, multilingual modeling, and transfer learning form the core of our approach in this area. More specifically, we find that the optimal approaches to this problem involve adapting existing ("pre-trained") models where possible, and employing *targeted* multilingual training to specialize models for a set of related or otherwise similar languages. The former leverages the fact that models trained at large scale have revolutionized the state of NLP over the past 6 years (so-called "foundation models" or "Large Language Models / LLMs"; Bommasani et al., 2022; Devlin et al., 2019; Conneau et al., 2020a; OpenAI et al., 2024, i.a.). The latter takes advantage of a middle ground between the "curse of multilinguality" encountered in massively multilingual settings on one hand, and the fact that low-resource languages seem to benefit from multilinguality, on the other.

In Chapter 2, we first introduce a novel modeling architecture for conducting unsupervised morphological segmentation, and then test its utility in multilingual and transfer-learning settings, applied to extremely low-resource languages. The model, termed a Masked Segmental Language Model (MSLM), can be trained on any language and learns to segment sentences into morphemes (the smallest meaning-bearing unit of language) without requiring segmentations provided by a human annotator. This is accomplished by treating a morphological segmentation as a latent path through a graph representing the sentence. The model then optimizes the marginal probability of all paths, and the output segmentation corresponds to the path with the highest probability. The ability to segment text into morphemes represents a critical first step to building useful tools for under-resourced languages, many of which are characterized by complex morphological systems not present in languages like English.

We demonstrate that a trained MSLM can be transferred to new languages with few or no training examples. When trained jointly on ten languages of Central and South America, the model achieves strong morpheme segmentation performance when transferred to K'iche' (Mayan), even with few or no training sentences in the new language. This demonstrates that multilingual training and transfer learning can be powerful techniques to bootstrap computational tools for critically under-resourced languages. It also underscores that neither the large model sizes nor large training sets typically employed to conduct multilingual machine learning are always necessary. The work in this chapter is published in *Proceedings of the 19th SIGMORPHON Workshop on Computational Phonetics, Phonology, and Morphology* (Downey et al., 2022b)[3] as well as *Proceedings of the 60th Meeting of Association for Computational Linguistics* (Downey et al., 2022a).[4]

Chapter 3 compares a range of methods for specializing the vocabulary of a multilingual "foundation" model for a specific language or languages. Such models are often pre-trained on ∼50-100 languages, and feature a vocabulary that is optimized for wide language coverage, which tends to ineffectively tokenize (segment) under-resourced languages (Ács, 2019; Rust et al., 2021). While it is common to adapt such models by simply continuing language

[3]Authors: C.M. Downey, Fei Xia, Gina-Anne Levow, Shane Steinert-Threlkeld

[4]Authors: C.M. Downey, Shannon Drizin, Levon Haroutunian, Shivin Thukral

model training on the language(s) of interest, retaining a vocabulary that covers unused languages incurs significant computational waste, and does not solve the problem of ineffective tokenization, since the vocabulary remains unchanged. For these reasons, it is desirable to replace the wide-coverage vocabulary with one that only covers the target language(s).

Vocabulary replacement hinges on formulating vector representations (embeddings) for any new vocabulary items that are introduced. We compare a range of new and previously proposed methods, from the simple baseline of randomly initializing new embeddings, to techniques that leverage vector-space similarity between tokens in the new and old vocabularies. Our experiments show that simple heuristics we propose based on the distribution of script (orthography) and word-internal position in the original vector space rival more complicated methods like the FOCUS algorithm, proposed elsewhere in the literature (Dobler and de Melo, 2023). Perhaps more importantly, we show that models adapted with a compact, specialized vocabulary perform as well or better than the original foundation model, while reducing model size and computational training cost by about 60%. This work is published in *Proceedings of the 3rd Workshop on Multilingual Representation Learning* (Downey et al., 2023).[5]

Finally, in Chapter 4, we systematically analyze the best approach for adapting a multilingual foundation model to a language family. We use the Uralic family as a test case, since it consists of many under-resourced languages that are not well-covered by current models, as well as a few relatively high-resource languages that may buoy performance for the rest by virtue of linguistic similarity. Importantly, it also has high-quality task evaluation data for a wide range of languages, allowing us to robustly test the utility of the adapted model. Adapting to a family follows the principle of targeted multilinguality by allowing very low-resource languages to pool data into a single model, while not incurring the "curse of multilinguality" — associated with covering too many languages or languages that are too dissimilar. Indeed, we show that Uralic-adapted models soundly outperform both the original "massively multilingual" model and models that are adapted to individual Uralic languages.

---

[5]Authors: C.M. Downey, Terra Blevins, Nora Goldfine, Shane Steinert-Threlkeld

In addition to showing that our family-wise adaptation presents significant advantages over multilingual and monolingual baselines, we conduct a statistical analysis of adaptation parameters and their resulting effect on model performance. This analysis reveals that although both training time (number of steps) and size of the adapted vocabulary positively contribute to model performance, training for longer is significantly more effective than choosing a larger vocabulary. We also show that low-resource languages can be aggressively over-sampled (i.e. repeated) during adaptation to the significant benefit of these languages, without significantly degrading performance in high-resource languages. This goes against conventional wisdom in multilingual modeling that over-sampling some languages will necessarily come at the cost of performance in the languages that are under-sampled to compensate.[6]

To conclude, we will briefly overview the most significant results from these studies and discuss their implications for the future of low-resource and multilingual NLP as a whole. We argue that the most promising avenues for rapidly expanding NLP advances to the world's languages include 1) leveraging useful, language-general components of pre-trained foundation models, rather than training "from scratch" for each new language, 2) substituting or substantially adapting components of pre-trained models that are not optimized for the language(s) in question, and 3) employing targeted multilingual modeling to leverage the advantages of data pooling and model-sharing for under-resourced languages, while avoiding the model capacity problems and language interference inherent in "massively multilingual" paradigms.

---

[6]This work is currently in preparation for submission to *Empirical Methods in Natural Language Processing*. Authors: C.M. Downey, Terra Blevins, Dhwani Serai, Dwija Parikh, Shane Steinert-Threlkeld

Chapter 2

UNSUPERVISED MULTILINGUAL PRE-TRAINING AND TRANSFER FOR MORPHOLOGICAL SEGMENTATION

Overview

This chapter first introduces a novel unsupervised model for conducting morphological segmentation, and then tests how this model can be used in tandem with multilingual training to facilitate transfer to extremely data-scarce languages. Significantly, we find that even relatively small models trained on a collection of languages that are low-resource but typologically similar to the target show non-trivial segmentation performance, even with few or no examples of the target language to train on. This goes against conventional wisdom that multilingual training necessarily involves large models or high-resource languages to anchor performance.

2.1 Introduction

Unsupervised sequence segmentation (at the word, morpheme, and phone level) has long been an area of interest in languages without whitespace-delimited orthography (e.g. Chinese, Uchiumi et al., 2015; Sun and Deng, 2018), morphologically complex languages without rule-based morphological analyzers (Creutz and Lagus, 2002), and automatically phone-transcribed speech data (Goldwater et al., 2009; Lane et al., 2021). It has been particularly important for lower-resource languages in which there is little or no gold-standard data on which to train supervised models (Joshi et al., 2020b).

In modern neural end-to-end systems, unsupervised segmentation is usually performed via information-theoretic algorithms such as BPE (Sennrich et al., 2016) and SentencePiece (Kudo and Richardson, 2018). However, the segmentations they produce are largely non-sensical to humans (Park et al., 2021). The motivating tasks listed above instead require unsupervised approaches that correlate more closely with human judgements of the boundaries of linguistic

units. For example, in a human-in-the-loop framework such as the *sparse transcription* proposed by Bird (2020), lexical items are automatically proposed to native speakers for confirmation, and it is important that these candidates be (close to) sensical, recognizable pieces of language.

In this work, we investigate the utility of recent models that have been developed to conduct unsupervised surface morpheme segmentation as a byproduct of a language modeling objective (Segmental Language Models, Kawakami et al., 2019, i.a.). The key idea is that recent breakthroughs in crosslingual language modeling and transfer learning (Conneau and Lample, 2019; Artetxe et al., 2020, i.a.) can be leveraged to facilitate transferring unsupervised segmentation performance to a new target language, using these types of language models.

First, we build upon a recent series of models termed Segmental Language Models (SLMs) by introducing a novel architecture variant: the Masked SLM (MSLM). This architecture is designed to leverage an entire sentence as context with which to make segmentation choices. In addition, its Transformer-based encoder allows scaling to deeper networks with less computational latency than its RNN-based predecessors. Secondly, we investigate the effectiveness of multilingual pre-training for a MSLM when applied to a low-resource target. We pre-train our model on the ten Indigenous languages of the 2021 AmericasNLP shared task dataset (Mager et al., 2021), and apply it to another low-resource, Indigenous, and morphologically complex language of Central America: K'iche' (quc), which is genealogically unrelated to the pre-training languages (Campbell et al., 1986).

We hypothesize that multilingual pre-training on similar, possibly contact-related languages, will outperform both a monolingual baseline trained from scratch as well as a model pre-trained on a single language (Quechua) with the same amount of pre-training data. We also expect that the pre-trained models will perform increasingly better than the monolingual baseline the smaller the target corpus is.

Indeed, our experiments show that a pre-trained multilingual model provides stable performance across all dataset sizes and far exceeds the monolingual baseline at low-to-medium target sizes. We additionally show that the multilingual model achieves a zero-shot segmentation performance of 20.6 F1 on the K'iche' data, where the monolingual baseline

has a score of zero. These results suggest that transferring from a multilingual model can greatly assist unsupervised segmentation in very low-resource languages, even those that are morphologically rich. The results also provide evidence for the idea that transfer from multilingual models works at a more moderate scale than is typical for recent crosslingual models (3.15 million parameters for our models).

In the following section, we overview work relating to unsupervised segmentation, crosslingual pre-training, and transfer-learning (§ 2.2). We then introduce the multilingual data used in our experiments, and the additional pre-processing we performed to prepare the data for pre-training (§ 2.3). Next we provide a brief overview of the type of Segmental Language Model used in our experiments, as well as our multilingual pre-training process (§ 2.4). After this, we describe our experimental process applying the pre-trained and from-scratch models to varying target data sizes (§ 2.5). Finally, we discuss the results of our experiments and their significance for low-resource pipelines, both within unsupervised segmentation and for other NLP tasks more generally (§ 2.6 and 2.7).

2.2 Related Work

2.2.1 Unsupervised Segmentation

Segmentation Techniques and SLM Precursors An early application of machine learning to unsupervised segmentation is Elman (1990), who shows that temporal surprisal peaks in RNNs provide a heuristic for inferring word boundaries. Subsequently, Minimum Description Length (MDL: Rissanen, 1989) was widely used. The MDL model family underlies well-known segmentation tools such as *Morfessor* (Creutz and Lagus, 2002) and other notable works (de Marcken, 1996; Goldsmith, 2001).

More recently, Bayesian models have proved some of the most accurate in their ability to model word boundaries. Some of the best examples are Hierarchical Dirichlet Processes (Teh et al., 2006), e.g. those applied to natural language by Goldwater et al. (2009), as well as Nested Pitman-Yor (Mochihashi et al., 2009; Uchiumi et al., 2015). However, Kawakami et al. (2019) note most of these do not adequately account for long-range dependencies in the same capacity as modern neural LMs.

Segmental Language Models follow a variety of recurrent models proposed for finding hierarchical structure in sequential data. Influential among these are Connectionist Temporal Classification (Graves et al., 2006), Sleep-Wake Networks (Wang et al., 2017), Segmental RNNs (Kong et al., 2016), and Hierarchical Multiscale Recurrent Neural Networks (Chung et al., 2017).

In addition, SLMs draw heavily from character and open-vocabulary language models. For example, Kawakami et al. (2017) and Mielke and Eisner (2019) present open-vocabulary language models in which words are represented either as atomic lexical units, or built out of characters. While the hierarchical nature and dual-generation strategy of these models did influence SLMs (Kawakami et al., 2019), both assume that word boundaries are available during training, and use them to form word embeddings from characters on-line. In contrast, SLMs usually assume no word boundary information is available in training.

Segmental Language Models § 2.4 has a more technical description of SLMs; here we give a short overview of related work. The term Segmental Language Model seems to be jointly due to Sun and Deng (2018) and Kawakami et al. (2019). Sun and Deng (2018) demonstrate strong results for Chinese Word Segmentation using an LSTM-based SLM and greedy decoding, competitive with and sometimes exceeding state of the art for the time.

Kawakami et al. (2019) use LSTM-based SLMs in a strictly unsupervised setting in which the model is only tuned to optimize language-modeling performance on the validation set, and is not tuned on segmentation quality. Here they report that "vanilla" SLMs give sub-par segmentations unless combined with one or more regularization techniques, including a character n-gram "lexicon" and length regularization.

Finally, Wang et al. (2021) introduce a bidirectional SLM based on a Bi-LSTM. They show improved results over the unidirectional SLM of Sun and Deng (2018), test over more supervision settings, and include novel methods for combining decoding decisions over the forward and backward directions. This proposed model is most similar to our own, though our transformer-based SLMs utilize a bidirectional context in a qualitatively different way, and do not require an additional layer to capture the reverse context.

2.2.2 Crosslingual and Transfer Learning

Crosslingual modeling and training has been an especially active area of research following the introduction of language-general encoder-decoders in neural machine translation, offering the possibility of zero-shot translation (i.e. translation for language pairs not seen during training; Ha et al., 2016; Johnson et al., 2017).

The arrival of crosslingual language model pre-training (XLM, Conneau and Lample, 2019) further demonstrates that large models pre-trained on multiple languages yield state-of-the-art performance across an abundance of multilingual tasks including zero-shot text classification (e.g. XNLI, Conneau et al., 2018), and that pre-trained transformer encoders provide great initializations for MT systems and language models in very low-resource languages.

Since XLM, numerous studies have attempted to single out which components of crosslingual training contribute to transferability from one language to another (e.g. Conneau et al., 2020b). Others have questioned the importance of multilingual training, and have instead proposed that even monolingual pre-training can provide effective transfer to new languages (Artetxe et al., 2020). Though some like Lin et al. (2019) have tried to systematically study which aspects of pre-training languages/corpora enable effective transfer, in practice the choice is often driven by availability of data and other ad-hoc factors.

Currently, large crosslingual successors to XLM such as XLM-R (Conneau et al., 2020a), MASS (Song et al., 2019), mBART (Liu et al., 2020), and mT5 (Xue et al., 2021) have achieved major success, and are the starting point for a large portion of multilingual NLP systems. These models all rely on an enormous amount of parameters and pre-training data, the bulk of which comes from very high-resource languages. In contrast, in this work we assess whether multilingual pre-training on a suite of very low-resource languages, which combine to yield a moderate amount of unlabeled data, can provide good transfer to similar languages which are also very low-resource.

2.3 Data and Pre-processing

We draw data from three main datasets. We use the AmericasNLP 2021 open task dataset (Mager et al., 2021) to pre-train our multilingual models. The multilingual dataset from Kann et al. (2018) serves as segmentation validation data for our pre-training process in these languages. Finally, data from Tyers and Henderson (2021) is used as the training set for our experiments transferring to K'iche', and Richardson and Tyers (2021) provides the validation and test data for these experiments.

AmericasNLP 2021 The AmericasNLP data consists of train and validation files for ten low-resource Indigenous languages of Central and South America: Asháninka (cni), Aymara (aym), Bribri (bzd), Guaraní (gug), Hñähñu (oto), Nahuatl (nah), Quechua (quy), Rarámuri (tar), Shipibo Konibo (shp), and Wixarika (hch). For each language, AmericasNLP also includes parallel Spanish sets, which we do not use. The data was originally curated for the AmericasNLP 2021 shared task on low-resource machine translation. (Mager et al., 2021).[1]

We augment the Asháninka and Shipibo-Konibo training sets with additional available monolingual data from Bustamante et al. (2020),[2] which is linked in the official AmericasNLP repository. We add both the training and validation data from this corpus to the *training* set of our splits.

To pre-process for a multilingual language modeling setting, we first remove lines that contain urls, copyright boilerplate, or that contain no alphabetic characters. We also split lines that are longer than 2000 characters into sentences/clauses where evident. Because we use the Nahuatl and Wixarika data from Kann et al. (2018) as validation data, we remove any overlapping lines from the AmericasNLP set. We create a combined train file as the concatenation of the training data from each of the ten languages, as well as a combined validation file likewise. All pre-processing scripts are found in our project repository.

Because the original ratio of Quechua training data is so high compared to all other languages (Figure 2.1), we downsample it to 2^{15} examples, the closest order of magnitude to

[1] `https://github.com/AmericasNLP/americasnlp2021`

[2] `https://github.com/iapucp/multilingual-data-peru`

the next-largest training set. A plot of the balanced (final) composition of our AmericasNLP train and validation sets is seen in Figure 2.2.

To compare the effect of multilingual and monolingual pre-training, we also pre-train a model on Quechua alone, since it has by far the most data (Figure 2.1). However, the full Quechua training set has about 50k fewer lines than our balanced AmericasNLP set (Figure 2.2). To create a fair comparison between multilingual and monolingual pre-training, we additionally create a downsampled version of the AmericasNLP set of equal size to the Quechua data (120,145 lines). The detailed composition of our data is available in Appendix A.1.

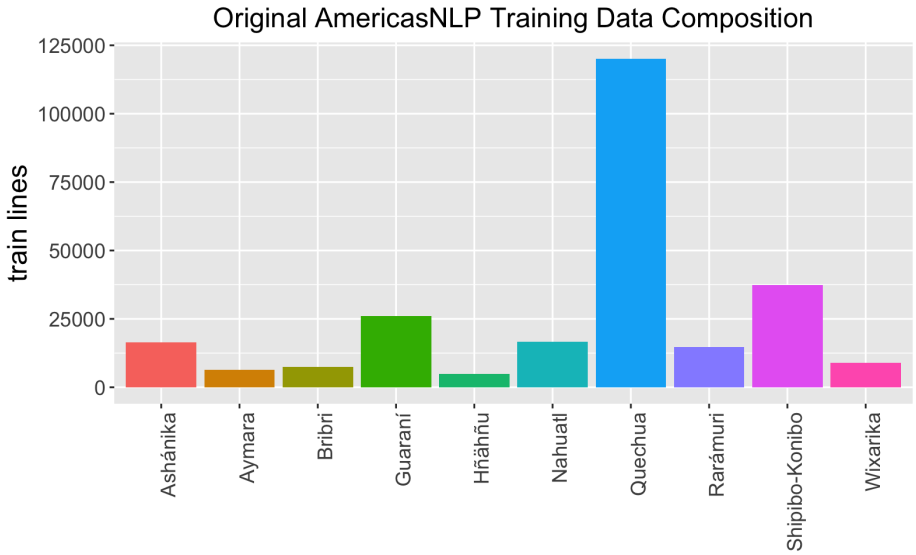


Figure 2.1: Original (imbalanced) language composition of the AmericasNLP training set

Kann et al (2018) The data from Kann et al. (2018) — originally curated for a segmentation task on polysynthetic low-resource languages — contains morphologically segmented sentences for Nahuatl and Wixarika. We use these examples as validation data for segmentation quality during the pre-training process. We clean this data in the same manner as the AmericasNLP sets.

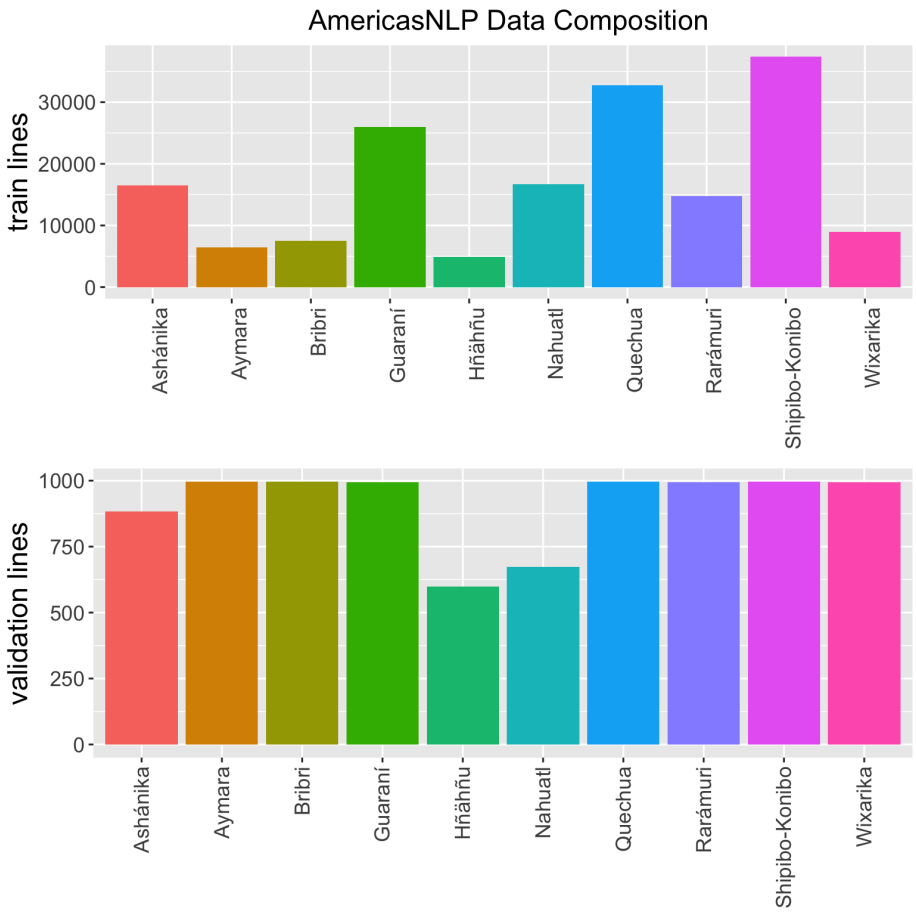


Figure 2.2: Final language composition of our AmericasNLP splits after downsampling Quechua

K'iche' data The K'iche' data used in our study was curated by Tyers and Henderson (2021). The raw (non-gold-segmented) data, used as the training set in our transfer experiments, comes from a section of this data web-scraped by the Crúbadán project (Scannell, 2007). This data is relatively noisy, so we clean it by removing lines with urls or lines where more than half of the characters are non-alphabetic. We also remove duplicate lines. The final data consists of 47,729 examples and is used as our full-size training set for K'iche'. Our experiments involve testing transfer at different resource levels, so we also create smaller training sets by downsampling the original to lower orders of magnitude.

For evaluating segmentation performance on K'iche', we use the segmented sentences from Richardson and Tyers (2021),[3] which were created for a shared task on morphological segmentation. These segmentations were created by a hand-crafted FST, then manually disambiguated. Because gold-segmented sentences are so rare, we concatenate the original train/validation/test splits and then split them in half into final validation and test sets.

2.4 Model and Pre-training

Recurrent SLMs A schematic of the original Recurrent SLM can be found in Figure 2.3. Within an SLM, a sequence of symbols or time-steps $\mathbf{x}$ can further be modeled as a sequence of segments $\underline{\mathbf{y}}$, which are themselves sequences of the input time-steps, such that the concatenation of segments $\pi(\underline{\mathbf{y}}) = \mathbf{x}$.

SLMs are broken into two levels: a context encoder and a segment decoder. The segment decoder estimates the probability of the j^{th} character in the segment starting at index i, y^i_j, as:

$$p(y^i_j | y^i_{0:j}, x_{0:i}) = Decoder(h^i_{j-1}, y^i_{j-1})$$

where the indices for $x_{i:j}$ are $[i, j)$. The context encoder encodes information about the input sequence up to index i. The hidden encoding h_i is

$$h_i = Encoder(h_{i-1}, x_i)$$

Finally, the context encoder "feeds" the segment decoder: the initial character of a segment beginning at i is decoded using (transformations of) the encoded context as initial states ($g_h(x)$ and $g_{start}(x)$ are single feed-forward layers):

$$p(y^i_0 | x_{0:i}) = Decoder(h^i_\emptyset, start^i)$$
$$h^i_\emptyset = g_h(h_{i-1})$$
$$start^i = g_{start}(h_{i-1})$$

For inference, the probability of a segment $\mathbf{y}_{i:i+k}$ (starting at index i and of length k) is modeled as the log probability of generating $\mathbf{y}_{i:i+k}$ with the segment decoder given the left

[3] `https://github.com/ftyers/global-classroom`

context $\pi(\underline{\mathbf{y}}_{0:i}) = x_{0:i}$. Note that the probability of a segment is **not** conditioned on other segments / segmentation choice, but only on the unsegmented input timeseries. Thus, the probability of the segment is

$$p(y_0^i|h_\emptyset^i, start^i) \prod_{j=1}^{k} p(y_j^i|h_{j-1}^i, y_{j-1}^i)$$

where y_k^i is the end-of-segment symbol.

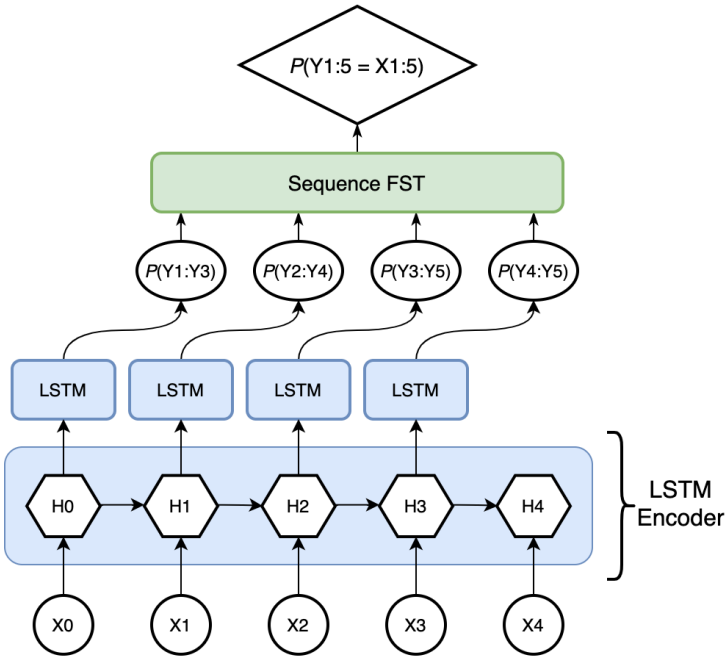


Figure 2.3: Recurrent Segmental Language Model

The probability of a sentence is thus modeled as the marginal probability over all possible segmentations of the input, as in equation (2.1) below (where $Z(|\mathbf{x}|)$ is the set of all possible segmentations of an input $\mathbf{x}$). However, since there are $2^{|\mathbf{x}|-1}$ possible segmentations, directly marginalizing is intractable. Instead, dynamic programming over a forward-pass lattice can

be used to recursively compute the marginal as in (2.2) given the base condition that $\alpha_0 = 1$. The maximum-probability segmentation can then be read off of the backpointer-augmented lattice through Viterbi decoding. Though a linguistic interpretation of this formulation is not straightforward, Kawakami et al. (2019) and Sun and Deng (2018) demonstrate an empirical correlation with human judgements of word boundaries in Chinese and English.

$$p(\mathbf{x}) = \sum_{\mathbf{z} \in Z(|\mathbf{x}|)} \prod_i p(\mathbf{y}_{i:i+z_i}) \tag{2.1}$$

$$p(\mathbf{x}_{0:i}) = \alpha_i = \sum_{k=1}^{L} p(\mathbf{y}_{i-k:i} | \mathbf{x}_{0:i-k}) \alpha_{i-k} \tag{2.2}$$

New Model: Masked SLM We propose a Masked Segmental Language Model, which leverages a non-directional transformer as the context encoder. This reflects recent advances in bidirectional (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005; Peters et al., 2018) and adirectional language modeling (Devlin et al., 2019). Such modeling contexts are also psychologically plausible: Luce (1986) shows that in acoustic perception, most words need some following context to be recognizable.

A key difference between our model and standard Masked LMs like BERT is that the latter predict single tokens based on the rest, while for SLMs we must predict a *segment* of tokens based on all other tokens *outside the segment*. For instance, to predict the three-character segment starting at x_t, the modeled distribution is $p(\mathbf{x}_{t:t+3} | \mathbf{x}_{<t}, \mathbf{x}_{\geq t+3})$.

Some recent pre-training techniques for transformers, such as SpanBERT (Joshi et al., 2020a), MASS (Song et al., 2019), and BART (Lewis et al., 2020) mask out spans to be predicted. A key difference between our model and these approaches is that the pre-training data for large transformer models is usually large enough that only about 15% of training tokens are masked, while we need to estimate the generation probability for *every* possible segment of $\mathbf{x}$. Since the usual method for masking is to replace the masked token(s) with a special symbol, only one span can be predicted with each forward pass. However, each sequence contains $O(|\mathbf{x}|)$ possible segments, so replacing each one with a mask token and recovering it would require as many forward passes.

These design considerations motivate our **Segmental Transformer Encoder**, and the **Segmental Attention Mask** around which it is based. Each forward pass of the encoder generates an encoding for every possible start-position in $\mathbf{x}$, for a segment of up to length k. The encoding at timestep $t-1$ corresponds to every possible segment whose first timestep is at index t. Thus with maximum segment length of k and total sequence length n, the representation at each index $t-1$ encodes

$$p(\mathbf{x}_{t:t+1}, \mathbf{x}_{t:t+2}, ...\mathbf{x}_{t:t+k} | \mathbf{x}_{<t}, \mathbf{x}_{\geq t+k})$$

This encoder leverages an attention mask that conditions predictions only on indices outside the predicted segment. An example of this mask with $k = 3$ is shown in Figure 2.4. For max segment length k, the mask is given by:

$$\alpha_{i,j} = \begin{cases} -\infty & \text{if } 0 < j - i \leq k \\ 0 & \text{else} \end{cases}$$

This solution is similar to that of Shin et al. (2020), developed independently and concurrently with our work, which uses a custom attention mask to "autoencode" each position without needing a special mask token. One key difference is that their masking scheme is used to predict single tokens, rather than spans. In addition, their mask runs directly along the diagonal of the attention matrix, rather than being offset. This means that to preserve self-masking in the first layer, the Queries are the "pure" positional embeddings.

To prevent information leaking "from under the mask", our encoder uses a different configuration in its first layer than in subsequent layers. In the first layer, Queries, Keys, and Values are all learned from the original input embeddings. In subsequent layers, the Queries come from the hidden encodings output by the previous layer, while Keys and Values are learned directly from the original embeddings. If Queries and either Keys or Values both come from the previous layer, information can leak from positions that are supposed to be masked for a particular query position. Shin et al. (2020) come to a similar solution to preserve their auto-encoder masking. The encodings learned by the segmental encoder are then input to an SLM decoder in exactly the same way as previous models (Figure 2.5).

Finally, to add positional information to the encoder, we use static sinusoidal encodings (Vaswani et al., 2017) and additionally apply a linear mapping f to the concatenation of the

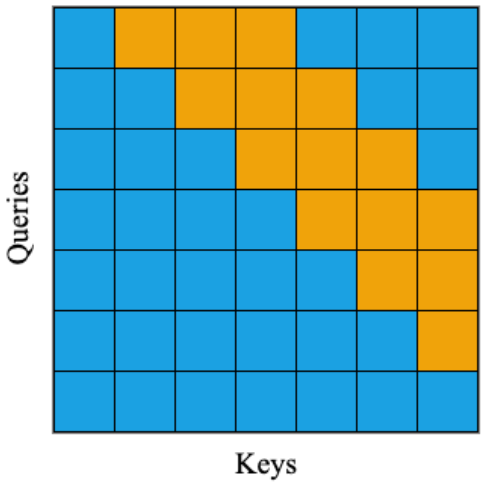


Figure 2.4: Segmental Attention Mask with segment-length (k) of 3. Blue squares are equal to 0, orange squares are equal to $-\infty$. This mask blocks the position encoding the segment in the Queries from attending to segment-internal positions in the Keys.

original and positional embeddings to learn the ratio at which to add the two together.

$$g = 1.0 + ReLU(f([embedding, position]))$$
$$embedding \leftarrow g * embedding + position$$

Pre-training Procedure In our experiments, we test the transferability of multilingual and monolingual pre-trained MSLMs. The multilingual models are trained on the AmericasNLP 2021 data (see § 2.3). Since SLMs operate on plain text, we can train the model directly on the multilingual concatenation of this data, and evaluate it by its language modeling performance on the concatenated validation data. As mentioned in § 2.3, we create two versions of the multilingual pre-trained model: one trained on the full AmericasNLP set (∼172k lines) and the other trained on the downsampled set, which is the same size as the Quechua training set (∼120k lines). We designate these models MULTI-PT$_{full}$ and MULTI-PT$_{down}$, respectively. Our pre-trained monolingual model is trained on the full

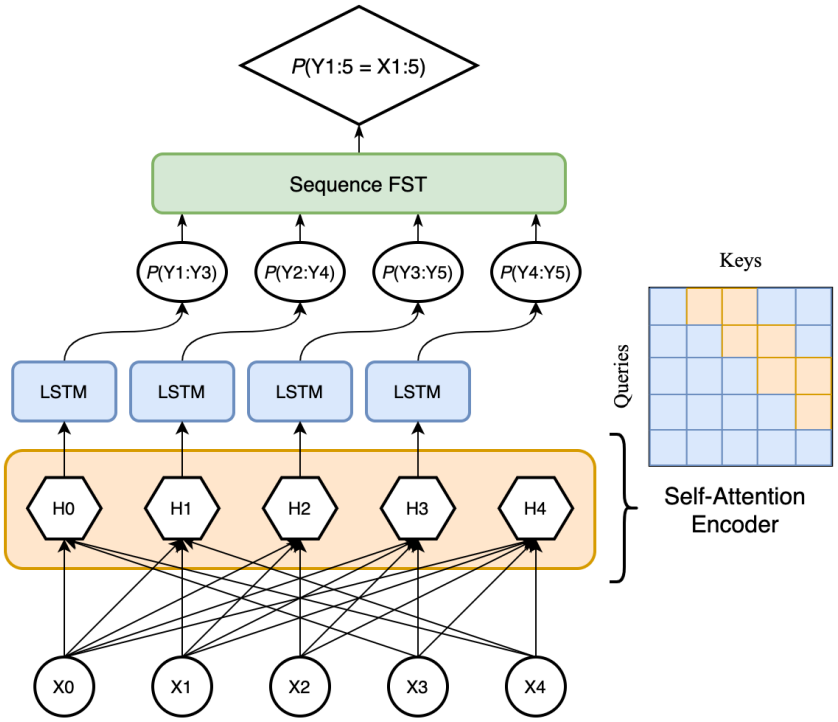


Figure 2.5: Masked Segmental Language Model, $k = 2$.

Quechua set (QUECHUA-PT).

Each model is an MSLM with four encoder layers, hidden size 256, feedforward size 512, and four attention heads. Character embeddings are initialized using Word2Vec (Mikolov et al., 2013) over the training data. The maximum segment size is set to 10. The best model is chosen as the one that minimizes the Bits Per Character (bpc) loss on the validation set. For further pre-training details, see Appendix A.2.

To evaluate the effect of pre-training on the segmentation quality for languages within the pre-training set, we also log Matthews Correlation Coefficient between the model output and gold-segmented secondary validation sets available in Nahuatl and Wixarika (Kann et al., 2018, see § 2.3). Figure 2.6 shows the unsupervised segmentation quality for Nahuatl and Wixarika almost monotonically increases during pre-training (MULTI-PT$_{full}$).

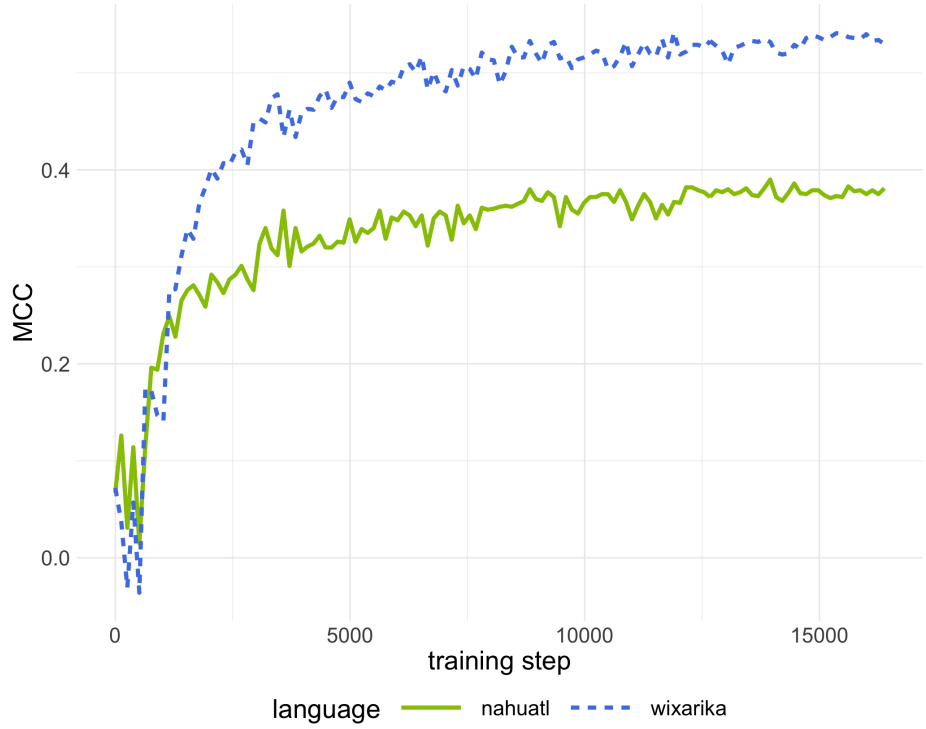


Figure 2.6: Plot of segmentation quality for Nahuatl and Wixarika during multilingual pre-training (measured by Matthews Correlation Coefficient with gold segmentation)

2.5 Experiments

We evaluate whether multilingual pre-training facilitates effective low-resource transfer learning for unsupervised segmentation. To do this, we pre-train SLMs on one or all of the AmericasNLP 2021 languages (Mager et al., 2021) and transfer it to a new target language: K'iche' (Tyers and Henderson, 2021). K'iche' is a morphologically rich Mayan language with several classes of inflectional prefixes and suffixes (Txchajchal Batz et al., 1996). An example sentence can be found in Table 2.1, which also shows our model's input and target output format.

As a baseline, we train a monolingual K'iche' model from scratch. We evaluate performance with respect to the size of the target training set, simulating varying degrees of

low-resource setting. To do this, we downsample the K'iche' training set to 8 smaller sizes, for 9 total: {256, 512, ... 2^{15}, 47.7k (full)}. For each size, we both train a monolingual baseline and fine-tune the pre-trained models we describe in § 2.4.[4]

Orthography	kinch'aw ruk' le nunan
Linguistic Segmentation	k-in-ch'aw r-uk' le nu-nan
Translation	"I speak with my mother"
Model Input	`kinch'awruk'lenunan`
Target Output	`k in ch'aw r uk' le nu nan`

Table 2.1: Example K'iche' sentence from Tyers and Henderson (2021). This sentence consists of multiple words, some of which consist of multiple morphemes. The model receives the sentence as an unsegmented stream of characters. The gold-standard (target) output is a sequence of morphemes (word and morpheme boundaries are treated the same, since the former is a subtype of the latter)

Architecture and Modeling All models are Masked Segmental Language Models (MSLMs) with the architecture described in § 2.4. The only difference is that the baseline model is initialized with a character vocabulary *only* covering the particular K'iche' training set (size-specific). The character vocabulary of the K'iche' data is a subset of the Americas-NLP vocabulary, so we are able to transfer the multilingual models without changing the embedding and output layers. The Quechua vocabulary is *not* a superset of the K'iche', so we add the missing characters to the Quechua model's embedding block *before* pre-training (these are randomly initialized). The character embeddings for the baseline are initialized using Word2Vec (Mikolov et al., 2013) on the training set (again, size-specific).

[4]All of the data and software required to run these experiments can be found at `https://github.com/cmdowney88/XLSLM`

Evaluation Metrics SLMs can be trained in either a fully unsupervised or "lightly" supervised manner (Downey et al., 2022b). In the former case, only the language modeling loss (Bits Per Character, bpc) is used to pick parameters and checkpoints. In the latter, the segmentation quality on gold-segmented validation data can be considered. Though our validation set is gold-segmented, we pick the best parameters and checkpoints based on bpc only, simulating the unsupervised case. However, to monitor the change in segmentation quality during training, we also use Matthews Correlation Coefficient (MCC). This measure frames segmentation as a character-wise binary classification task (i.e. boundary vs. no boundary), and measures correlation with the gold segmentation.

To make our results comparable with the wider word-segmentation literature, we use the scoring script from the SIGHAN Chinese Word Segmentation Bakeoff (Emerson, 2005) for our final segmentation F1. For each model and target size, we choose the best checkpoint (by bpc), apply the model to the combined validation and test set, and use the SIGHAN script to score the output.

For comparison to the Chinese Word Segmentation and speech literature, any whitespace segmentation in the validation/test data is discarded before it is fed to the model. However, SLMs can also be trained to treat spaces like any other character, and thus could be able to take advantage of existing segmentation in the input. We leave this for future work.

Parameters and Trials For our training procedure (both training the baseline from scratch and fine-tuning the pre-trained models) we tune hyperparameters on three of the nine dataset sizes (256, 2048, and full) and choose the optimal parameters by bpc. For each of the other sizes, we directly apply the chosen parameters from the tuned dataset of the closest size (on a log scale). We tune over five learning rates and three encoder dropout values. As in pre-training, we set the maximum segment length to 10. For more details on our training procedure, see Appendix A.2.

2.6 Results

The results of our K'iche' transfer experiments at various target sizes can be found in Table 2.2. In general, the (full) pre-trained multilingual model (MULTI-PT$_{full}$) demonstrates

good performance across dataset sizes, with the lowest segmentation performance (20.6 F1) being in the zero-shot case and the highest (40.7) achieved on 2^{14} examples. The monolingual baseline outperforms MULTI-PT$_{full}$ at the two largest target sizes, as well as at size 4096 (achieving the best overall F1 of 44.8), but performs very poorly under 2048 examples, and has no zero-shot ability (unsurprisingly, since it is a random initialization).

Interestingly, other than in the zero-shot case, QUECHUA-PT and the comparable MULTI-PT$_{down}$ perform very similarly to each other. However, the zero-shot transferability of MULTI-PT$_{down}$ is almost twice that of the model trained on Quechua only. MULTI-PT$_{full}$ exceeds both MULTI-PT$_{down}$ and QUECHUA-PT by a wide margin in every setting. Finally, all models show increasing performance until about size 4096, after which more target examples don't provide a large increase in segmentation quality.

Model	Target Language Segmentation F1									
	0	256*	512	1024	2048*	4096	8192	2^{14}	2^{15}	47,729 (full)*
MULTI-PT$_{full}$	**20.6**	**34.0**	**37.4**	**37.4**	38.2	40.5	**38.6**	**40.7**	38.9	38.2
MULTI-PT$_{down}$	15.0	25.1	25.7	29.3	32.5	33.2	33.3	31.5	33.6	31.9
QUECHUA-PT	7.6	29.9	31.0	30.4	30.7	31.0	29.9	33.6	31.8	33.3
MONOLINGUAL	0.002	4.0	3.3	10.3	**39.2***	**44.8**	29.4	39.5	**44.1**	**43.2**

Table 2.2: Segmentation quality on the combined validation and test set for each model, at each target training set size. Star indicates size at which hyperparameter tuning is conducted. For tuned sizes, showing only the performance of the model with the best bpc. *See Table 2.3: the best baseline trial achieved slightly better performance than MULTI-PT$_{full}$, but the former is far more sensitive to variation due to hyperparameters at this size

Interpretation These results show that MULTI-PT$_{full}$ provides consistent performance across target sizes as small as 512 examples. Even for size 256, there is only a 9% (relative) drop in quality from the next-largest size. Further, the pre-trained model's zero-shot performance is impressive given the baseline is effectively 0 F1.

Model	Target Language Segmentation F1		
	256*	2048*	47,729 (full)*
MULTI-PT$_{full}$	**34.2** ± **0.6** (1.8%)	**38.1** ± **0.4** (1.0%)	39.4 ± 1.1 (2.8%)
MULTI-PT$_{down}$	25.7 ± 0.6 (2.3%)	30.5 ± 2.3 (7.5%)	31.7 ± 0.6 (1.9%)
QUECHUA-PT	30.1 ± 0.2 (0.7%)	31.4 ± 0.6 (1.9%)	32.7 ± 0.7 (2.1%)
MONOLINGUAL	4.2 ± 0.5 (11.9%)	36.5 ± 6.8 (18.6%)	**44.7** ± **2.0** (4.5%)

Table 2.3: Variation of segmentation quality across the best four hyperparameter combinations for a single size (by bpc; mean ± standard deviation (stdev ÷ mean); models ranked by mean minus stdev)

On the other hand, the performance of the monolingual baseline at larger sizes seems to suggest that given enough target data, it is better to train a model devoted to the target language only. This is consistent with previous results on general language modeling (Wu and Dredze, 2020; Conneau et al., 2020a). However, it should also be noted that MULTI-PT$_{full}$ never trails the baseline by more than 5.2 F1.

One less-intuitive result is the dip in the baseline's performance at sizes 8192 and 2^{14}. We believe this discrepancy may be partly explainable by sensitivity to hyperparameters in the baseline. Though the best baseline trial at size 2048 exceeds MULTI-PT$_{full}$ by a small margin, the baseline shows large variation in performance across the top-four hyperparameter settings at this size, where MULTI-PT$_{full}$ actually performs better on average and much more consistently (Table 2.3). We thus believe the dip in performance for the baseline at sizes 8192 and 2^{14} may be due to an inability to extrapolate hyperparameters from other experimental settings.

2.7 Discussion

Standing of Hypotheses Within the framework of unsupervised segmentation, these results provide strong evidence that relevant linguistic patterns can be learned over a

collection of low-resource languages, and then transferred to a new language without much (or any) target training data. Further, it is shown that the target language need not be (phylogenetically) related to any of the pre-training languages, even though details of morphological structure are ultimately language-specific.

The hypothesis that multilingual pre-training yields increasing advantage over a from-scratch baseline at smaller target sizes is also strongly supported. This result is consistent with related work showing this to be a key advantage of the multilingual approach (Wu and Dredze, 2020).

The hypothesis that multilingual pre-training also yields better performance than monolingual pre-training given the same amount of data seems to receive mixed support from our experiments. On one hand, the comparable multilingual model has a clear advantage over the Quechua model in the zero-shot setting, and outperforms the latter in 5/10 settings more generally. However, because the Quechua data lacks several frequent K'iche' characters (and these embeddings remain randomly initialized), it is unclear how much of this advantage comes from the multilingual training *per-se*. Instead, the advantage may be due to the multilingual model's full coverage of the target vocabulary— an advantage which may disappear at larger target sizes. Further analysis of this hypothesis will require additional investigation.

Significance The above results, especially the strong zero-shot transferability of segmentation performance, suggest that the type of language model used here learns some abstract linguistic pattern(s) that are generalizable across languages, and even to new ones. It is possible that these generalizations could take the form of abstract stem/affix or word-order patterns, corresponding roughly to the lengths and order of morphosyntactic units. Because MSLMs operate on the character level (and in these languages orthographic characters mostly correspond to phones), it is also possible the model could recognize syllable structure in the data (the ordering of consonants and vowels in human languages is relatively constrained), and learn to segment on syllable boundaries.

It is also helpful to remember that we select the training suite and target language to have some characteristics in common that may help facilitate transfer. The AmericasNLP

languages are almost all morphologically rich, with many considered polysynthetic (Mager et al., 2021), a feature that K'iche' shares (Suárez, 1983). Further, all of the languages, including K'iche', are spoken in countries where either Spanish or Portuguese is the official language, and have very likely had close contact with these Iberian languages and borrowed lexical items. Finally, the target language family (Mayan) has also been shown to have close historical contact with the families of several of the AmericasNLP set (Nahuatl, Rarámuri, Wixarika, Hñähñu), forming a Linguistic Area or *Sprachbund* (Campbell et al., 1986).

It is possible that one or several of these shared characteristics facilitates the strong transfer shown here, in both our multilingual and monolingual pre-trained models. However, our current study does not conclusively show this to be the case. Lin et al. (2019) show that factors like linguistic similarity and geographic contact are often not as important for transfer success as non-linguistic features such as the raw size of the source dataset. Indeed, the fact that our Quechua pre-trained model performs similarly to the comparable multilingual model (at least at larger target sizes) suggests that the benefit to using MULTI-PT$_{full}$ could be interpreted as a combined advantage of pre-training data size and target vocabulary coverage.

The nuanced question of whether multilingual pre-training *itself* enables better transfer than monolingual pre-training requires more study. However, taking a more pragmatic point of view, multilingual training can be seen as a methodology to 1) acquire more data than is available from any one language and 2) ensure broader vocabulary overlap with the target language. Our character-based model is of course different from more common word- or subword-based approaches, but with these too, attaining pre-trained embeddings that cover a novel target language is an important step in cross-lingual transfer (Garcia et al., 2021; Conneau et al., 2020a; Artetxe et al., 2020, *inter alia*)

Future Work We believe some future studies would shed light on the nuances of segmentation transfer-learning. First, pre-training either multilingually or monolingually on languages that are *not* linguistically similar to the target language could help isolate the advantage given by pre-training on *any* language data (vs. similar language data).

Second, we have noted that monolingual pre-training on a language that does not have

near-full vocabulary coverage of the target language leaves some embeddings randomly initialized, yielding worse performance at small target sizes. Pre-training a model on a single language that happens to have near-complete vocabulary coverage of the target could give a better view of whether multilingual training intrinsically yields advantages, or whether monolingual training is disadvantaged mainly due to this lack of vocabulary coverage.

Finally, because no authors of this work have any training in the K'iche' language, we are unable to perform a linguistically-informed error analysis of our model's output (e.g. examining the types of words and morphemes which are erroneously (un)segmented, rather than calculating an overall precision and recall for the predicted and true morpheme boundaries, as we do in this study). However, we make all of our model outputs available in our public repository, so that future work may provide a more nuanced analysis of the types of errors unsupervised segmentation models are prone to make.

2.8 Conclusion

This study has shown that unsupervised sequence segmentation ability can be transferred via multilingual pre-training to a novel target language with little or no target data. The target language also need not be from the same family as a pre-training language for successful transfer. While training a monolingual model from scratch on large amounts of target data results in good segmentation quality, our experiments show that pre-trained models, especially multilingual ones, far exceed the baseline at small target sizes ($\leq$1024), and seem to be much more robust to hyperparameter variation at medium sizes (2048, 8192, 2^{14}).

One finding that may have broader implications is that pre-training can be conducted over a set of low-resource languages with some typological or geographic connection to the target, rather than over a crosslingual suite centered around high-resource languages like English and other European languages. Most modern crosslingual models have huge numbers of parameters (XLM has 570 million, mT5 has up to 13 billion, Xue et al., 2021), and are trained on enormous amounts of data, usually bolstered by hundreds of gigabytes in the highest-resource languages (Conneau et al., 2020a).

In contrast, our results suggest that effective transfer may be possible at smaller scales, by combining the data of low-resource languages and training moderately-sized, more targeted

pre-trained multilingual models (our model has 3.15 million parameters). Of course, this study can only support this possibility within the unsupervised segmentation task, so future work will be needed to investigate whether transfer to and from low-resource languages can be extended to other tasks.

Relation to Remaining Work

The following two chapters will focus on adaptation techniques for large, "massively multilingual" foundation models, rather than the compact models pre-trained in this chapter. Where the small multilingual model trained in this chapter lent itself to rapid adaptation to new languages, more sophisticated techniques will be introduced to specialize large multilingual models for specific target language(s). In particular, Chapter 3 will introduce methods to replace a massively multilingual vocabulary with one specialized for a narrower set of languages, and Chapter 4 will investigate important dynamics for adapting to a language family. However, both these chapters will follow the principles introduced in Chapter 2 of leveraging pre-trained models as a starting point for under-resourced languages, and of using targeted multilingual training to counter data sparsity in very under-resourced languages.

Chapter 3

COMPARING METHODS TO ADAPT MULTILINGUAL VOCABULARIES TO NEW LANGUAGES

Overview

This chapter compares a range of techniques for specializing the vocabulary of a pre-trained cross-lingual model for specific target languages. Unlike Chapter 2, the pre-trained model we leverage here is large and extensively pre-trained in a "massively multilingual" setting. It is also a general-purpose language model, rather than one specialized for segmentation. The vocabulary specialization investigated here is designed to mitigate the fact that the original massively multilingual vocabulary and tokenizer are usually inefficient and ineffective for under-resourced languages. In addition, retaining vocabulary representations for languages that go unused during adaptation incurs significant computational waste. The specialized vocabularies we initialize here are compact enough to reduce overall computation and model size by 60%. Our comparison of specialization techniques reveals that even simple heuristic-based techniques — leveraging relevant representations from the original model — are adequate to yield large modeling performance gains for under-resourced languages.

3.1 Introduction

For languages other than English and a handful of other very high-resource languages, pre-trained multilingual language models form the backbone of most current NLP systems. These models address the relative data scarcity in most non-English languages by pooling text data across many languages to train a single model that (in theory) covers all training languages (Devlin, 2019; Conneau and Lample, 2019; Conneau et al., 2020a; Liu et al., 2020; Scao et al., 2023, i.a.). These models often include language-agnostic tokenization and an increased vocabulary capacity over monolingual models (Conneau et al., 2020a).

However, Wu and Dredze (2020) show that these massively multilingual models still

underperform on lower-resource languages. Recent efforts to cover these languages instead pre-train models that are specialized to specific languages or language families (Ogueji et al., 2021; Ogunremi et al., 2023). These approaches nonetheless require training a new model from scratch and do not leverage transferable information in existing models.

Our study builds on a line of work which instead *adapts* a pre-trained cross-lingual model (such as XLM-R; Conneau et al., 2020a) to a single language, or a smaller set of languages. Language-Adaptive Pre-Training (LAPT)—continuing the MLM or CLM pre-training task on only the target language(s)—is a simple and strong baseline in this regard (Chau et al., 2020).

However, LAPT with no change to the cross-lingual vocabulary comes with considerable excess computational cost: when adapting to a single language or small subset of languages, only a small fraction of the cross-lingual vocabulary is used. The excess vocabulary still contributes to the computational cost on both the forward and backward pass, and embedding/output matrices often constitute a large fraction of the total trainable model parameters (for XLM-R-base, 192M / 278M $\approx$ 69% of parameters). Additionally, the information-theoretic tokenization modules for cross-lingual models are usually under-optimized for any given language, and especially low-resource languages (Ács, 2019; Conneau and Lample, 2019, i.a.)

For this reason, we propose several simple techniques to replace the large cross-lingual vocabulary of a pre-trained model with a compact, language-specific one during model specialization. Training a new SentencePiece or BPE tokenizer poses no special difficulties. However, re-initializing the embedding matrix for a new vocabulary, which will almost certainly introduce many new tokens lacking pre-trained embeddings, poses significant challenges. We compare several methods for such embedding re-initialization.

After reviewing related literature in § 3.2, we conduct a qualitative exploration of the pre-trained embedding space for a standard multilingual model: XLM-R (§ 3.3.1). This exploration informs our formalization of simple techniques to align new vocabulary embeddings with the pre-trained embedding distribution of our base model (§ 3.3.2). We then provide a systematic experimental comparison of the embedding re-initialization techniques we propose, plus the recently proposed FOCUS re-initialization method (Dobler and de Melo,

2023, § 3.4). Our experiments cover a wide selection of low- and mid-resource target languages (i.e. those that have the most to gain from language specialization).[1]

The results of our experiments (§ 3.5, 3.6) demonstrate the following: 1) Embedding-replacement techniques proposed in the monolingual model adaptation literature are inadequate for adapting multilingual models. 2) Replacing large cross-lingual vocabularies with smaller language-specific ones provides a computationally-efficient method to improve task performance in low-resource languages. 3) The simple re-initialization techniques we propose here, based on script-wise embedding sub-distributions, rival techniques such as FOCUS, which rely on model-driven semantic similarity.

3.2 Related Work

Pre-trained Model Adaptation Extensive work has proposed re-using and modifying pre-trained models for new settings in order to retain existing model knowledge and reduce pre-training costs. Gururangan et al. (2020) show that continued training on domain-specific data effectively adapts pre-trained models to new domains in both high- and low-resource settings. This approach is also used to adapt models to new languages (i.e. Language-Adaptive Pre-Training / LAPT; Chau et al., 2020).

Other approaches involve training new, language-specific adapter layers to augment a frozen monolingual (Artetxe et al., 2020) or multilingual encoder (Pfeiffer et al., 2020; Üstün et al., 2020; Faisal and Anastasopoulos, 2022). A comparison of these cross-lingual adaptation approaches (Ebrahimi and Kann, 2021) found that continued pre-training often outperforms more complex setups, even in low-resource settings. With this in mind, our experiments evaluate the success of models tuned for target languages with LAPT, starting from variable initializations depending on a choice of embedding adaptation technique.

Cross-lingual Vocabulary Adaptation A major limitation in adapting pre-trained models to new languages is the subword vocabulary, which often fails to cover an unseen script (Pfeiffer et al., 2021) or tokenizes target text inefficiently (Ács, 2019). Muller et al.

[1]The software used to run all experiments may be found at `https://github.com/cmdowney88/EmbeddingStructure`

(2021) demonstrate that script is an extremely important factor in predicting transfer success. Specifically, the pre-trained coverage of closely-related languages improves transfer, but only if the target language is written in the same script as its pre-trained relative.

One adaptation technique is to initialize new subword embeddings that cover the target language, e.g. by expanding the existing vocabulary with new tokens as necessary, then training the new (randomly initialized) embeddings (Chau et al., 2020; Wang et al., 2020a). When transferring a monolingual model to a new language, Artetxe et al. (2020) and de Vries and Nissim (2021) instead completely re-initialize the embedding matrix, corresponding to a new subword vocabulary. These embeddings are then trained into alignment with the pre-trained, frozen transformer encoder. We show that this technique is not successful when adapting a multilingual model (§ 3.5).

Other work reuses information in pre-trained embeddings rather than initializing new ones at random. This may include scaling up smaller embedding spaces from models trained on the target language (de Vries and Nissim, 2021; Ostendorff and Rehm, 2023) or copying embeddings from the original vocabulary where there is exact vocabulary overlap (Pfeiffer et al., 2021). When transferring to a target language written in a poorly-covered script, Muller et al. (2021) show that transliterating the target to the script of a well-covered relative can lead to significant performance gains, a result also noted in the low-resource machine translation literature (Neubig and Hu, 2018; Amrhein and Sennrich, 2020).

Finally, recent work has proposed more complex methods for mapping source embeddings onto semantically similar ones in the target space either through cross-lingually aligned static word embeddings (e.g. the WESCHEL method; Minixhofer et al., 2022) or with bilingual lexicons (Zeng et al., 2023). In concurrent work to ours, Dobler and de Melo (2023) extend WECHSEL with the FOCUS method to specialize multilingual vocabularies to a single language. Ostendorff and Rehm (2023) use a cross-lingual progressive transfer learning approach to combine information from the source embeddings and a smaller target language model to initialize higher-dimension target embeddings. Unlike earlier initialization methods and our proposed setup, these methods all require additional information outside the source model and often require significant additional compute. We compare one method from this family (FOCUS) to our proposed heuristic-based initialization schemes.

3.3 Vocabulary Replacement & Embedding Re-initialization

Research transferring monolingual models from one language to another (e.g. Artetxe et al., 2020; de Vries and Nissim, 2021), has shown that random re-initialization of embeddings +LAPT is sufficient. However, our experiments show that this technique performs poorly when transferring from a multilingual model (§ 3.5). For this reason, we propose several simple techniques for initializing new embeddings based on a qualitative exploration of the embedding space for XLM-R (§ 3.3.1), and include the more complex FOCUS technique, developed concurrently with our work, for comparison (Dobler and de Melo, 2023).

3.3.1 XLM-R Embedding-Space Analysis

To better understand the task of initializing new embeddings for a multilingual model, we explore the token-embedding space of XLM-R through PCA projection. Our hypothesis is that multilingual models do not process all languages homogeneously. This seems to be demonstrated in Figures 3.1a and 3.1b, where word embeddings are colored by their respective Unicode script block. We see that the highest-resource scripts in XLM-R (Common, Latin, and Cyrillic) have relatively divergent distributions, while others cluster closer together. This heterogeneity may help explain the finding from Muller et al. (2021) that pre-trained models do not transfer well to even closely-related target languages if the target script does not match that of the pre-trained relative.

Secondly, each script can be further divided into two sub-distributions, roughly corresponding to a shift in the second principal component. Figure 3.1c shows that this division corresponds to whether a token is word-initial or word-medial. To preserve whitespace information, SentencePiece tokens include a leading underscore to indicate tokens that should be preceded by a space (word-initial tokens).[2] Although the model does not have access to the internal makeup of its tokens, we hypothesize that it learns to discern which tokens can begin a word and which cannot.

Thus when proposing methods to initialize new embeddings for XLM-R, we hypothesize that initializing according script- and position-wise sub-distributions will help to align new

[2]E.g., "_the" and "the" are word-initial and word-medial tokens of the same character sequence.

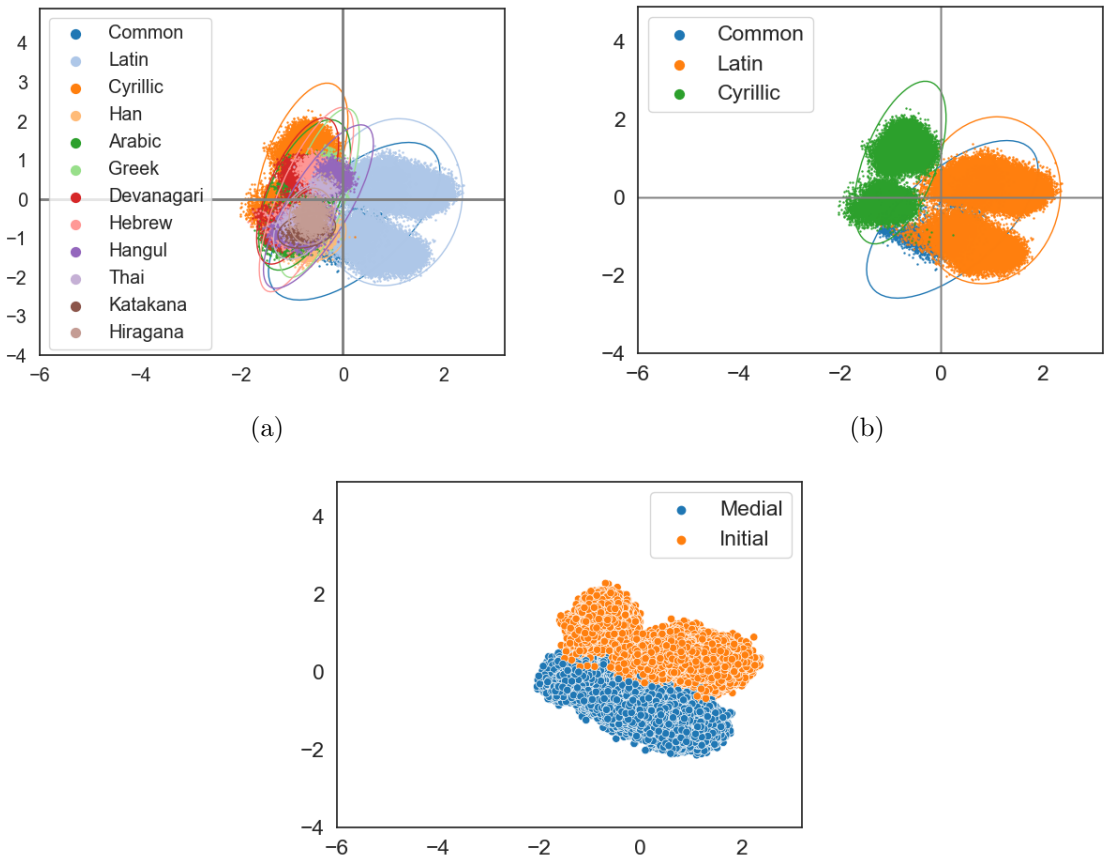


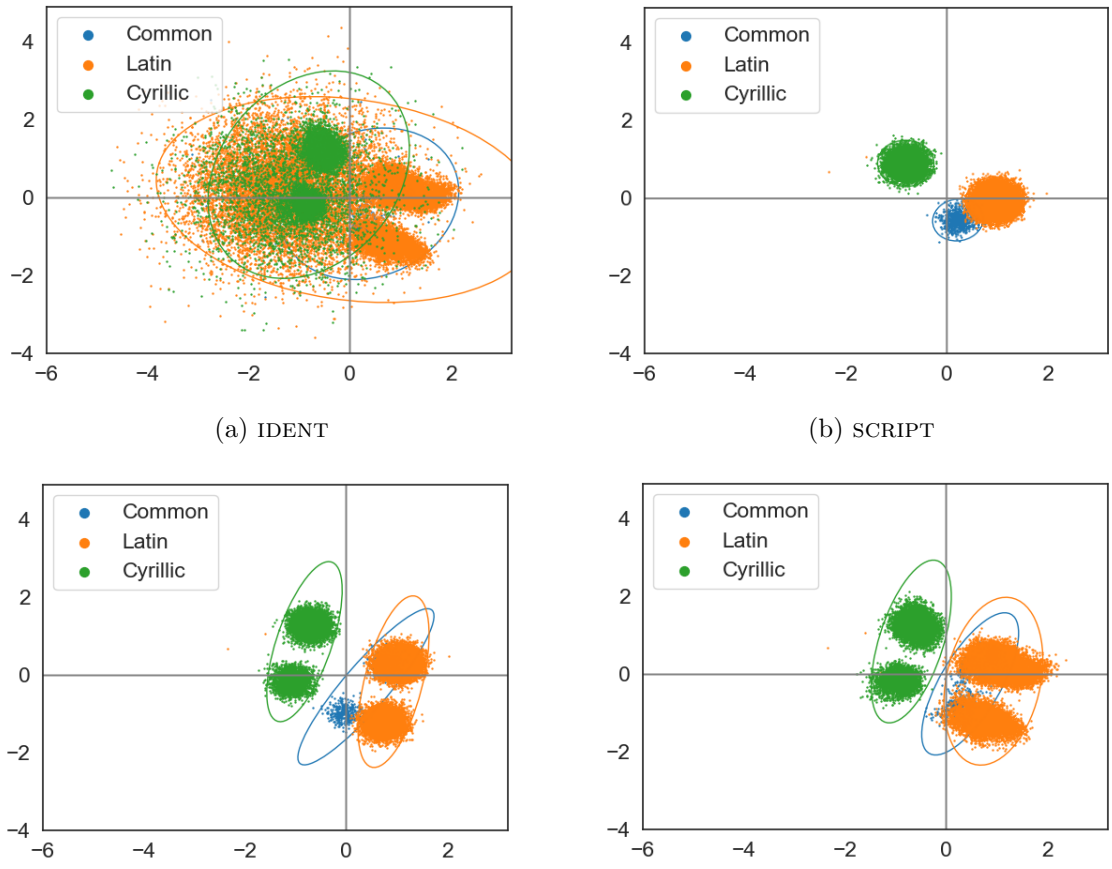
Figure 3.1: PCA visualizations of the embedding space for XLM-R. Subplots: (a) Distribution of embeddings for the 12 most common Unicode scripts. (b) Plot reduced to only Common, Latin, and Cyrillic scripts for simplicity. (c) Embeddings colored by whether the token begins a word (initial) or occurs in the middle of one (medial)

vocabulary items with the pre-trained embedding distribution.

3.3.2 Embedding Re-initialization Techniques

We now formalize simple techniques for embedding re-initialization based on our exploration of XLM-R's embedding space, as well as one recently proposed technique based on an

auxiliary embedding model (FOCUS). Figure 3.2 provides PCA visualizations of the re-initialized embeddings from each technique on a subword vocabulary specialized for languages of the Uralic family (we experiment with these languages in § 3.4). The visualization for these languages' respective scripts (Common, Latin, Cyrillic) in the base model can be found in Figure 3.1b for comparison.



(a) IDENT (b) SCRIPT

(c) SCRIPT+POSN (d) SCRIPT+POSN+IDENT

Figure 3.2: PCA visualizations embedding re-initialized using the heuristic techniques introduced in § 3.3.2

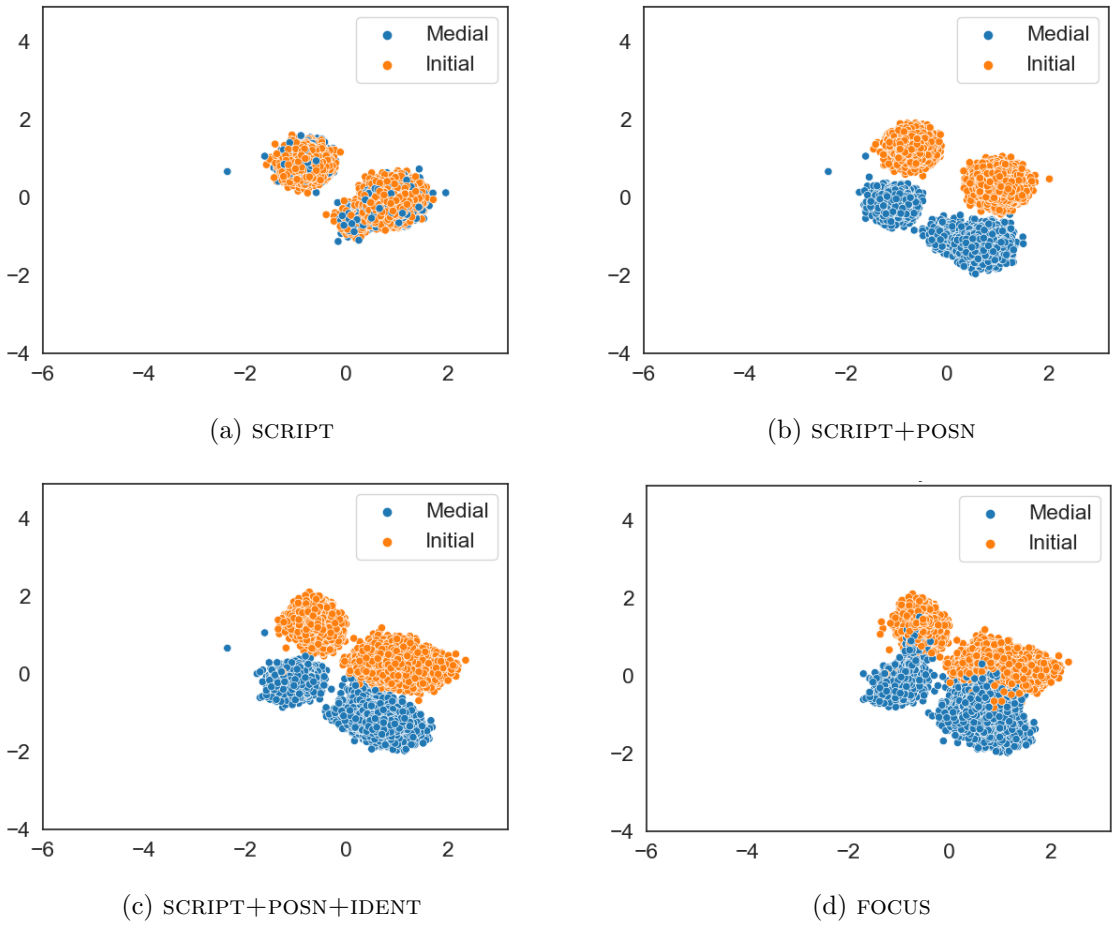
Re-initialization by Identity REINIT-IDENT first identifies tokens in the new vocabulary that exactly match a token in the original vocabulary, then sets the new embeddings of

shared tokens to be identical to those in the original embedding table (Figure 3.2a). This is a common approach to preserve information from the original model, even when the other embeddings are randomly re-initialized (e.g., Pfeiffer et al., 2021). When identity re-initialization is applied in conjunction with another technique (such as REINIT-SCRIPT), identity takes precedence.

Re-initialization by Script For REINIT-SCRIPT, all base XLM-R tokens are first categorized by Unicode block, as a stand-in for identifying the script/orthography. We then calculate the mean and standard deviation for each script in the original embedding space. Finally, new token embeddings for each script are distributed according to a Normal distribution with the corresponding mean and standard deviation (Figure 3.2b).

Re-initialization by Position REINIT-POSN is based on the observation that within each script, embeddings seem to cluster according their word-initial vs. word-medial status (Figure 3.1c). Similarly to REINIT-SCRIPT, we identify the mean and standard deviation of embeddings that belong to each category. Because positional status seems to be a sub-cluster within script clusters, we only use REINIT-POSN in combination with REINIT-SCRIPT. The mean and standard deviation for each (script, position) combination is calculated and new embeddings are initialized accordingly (Figure 3.2c).

Focus Re-initialization In addition to the heuristic-based methods introduced above, we investigate a pre-existing method for embedding transfer, termed FOCUS (Dobler and de Melo, 2023). FOCUS works by extrapolating from the embedding space of an existing model, like our heuristic methods, but further introduces an auxiliary embedding model trained on the new language(s). This auxiliary model (based on FastText; Bojanowski et al., 2017) is used to obtain similarity measures between the new vocabulary items. Embeddings corresponding to overlapping tokens in the new vocabulary keep their values from the source model (REINIT-IDENT). Completely new tokens are initialized as a weighted combination of the overlapping items, with weights obtained according to similarity in the auxiliary model.



(a) SCRIPT (b) SCRIPT+POSN

(c) SCRIPT+POSN+IDENT (d) FOCUS

Figure 3.3: PCA visualization of re-initialized embeddings with word-initial vs word-medial tokens highlighted. For REINIT-SCRIPT, position-wise clustering seen in the base XLM-R embeddings (Figure 3.3a) is not captured. REINIT-SCRIPT+POSN and REINIT-SCRIPT+POSN+IDENT show expected positional clustering. REINIT-FOCUS seems to allow slightly more positional overlap

Random Re-initialization Embeddings not initialized through the above methods are initialized according to a Standard Normal Distribution about the origin. This includes the non-overlapping tokens when REINIT-IDENT is applied on its own, and REINIT-RANDOM, where all embeddings are initialized this way.

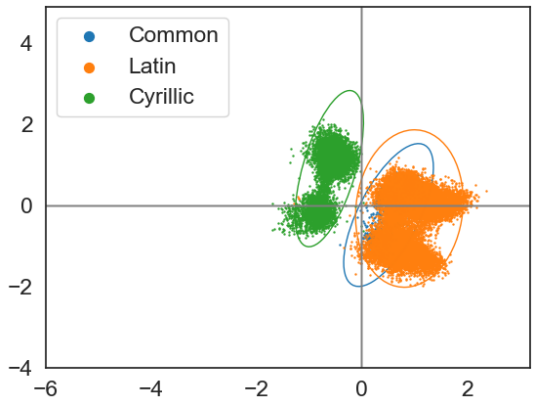


Figure 3.4: PCA: REINIT-FOCUS embeddings

Inspection of re-initialized embeddings Figures 3.2 and 3.4 show PCA visualizations for the re-initialization techniques described here. Figure 3.2a shows that while REINIT-IDENT captures some of the pre-trained embedding structure, a large number also remain randomly scattered throughout the space. REINIT-SCRIPT (3.2b) initializes all embeddings in a Normal distribution about the centroid for each script, but misses key embedding structure, such as the fact that each script has two position-wise sub-distributions. REINIT-SCRIPT+POSN (3.2c) takes these sub-distributions into account, forming six Normal clusters instead of three. Figure 3.3b verifies that these clusters capture the initial vs. medial token distinction. Finally, REINIT-SCRIPT+POSN+IDENT (3.2d) and FOCUS (3.4) give the closest emulation of the original XLM-R embedding structure (3.1b).

3.4 Experiments

In our experiments, we replace the large cross-lingual embedding matrix of XLM-R and re-initialize it for a new, language-specific vocabulary. We then conduct LAPT to specialize the model for the new language(s), and evaluate performance on downstream tasks. We consider both multilingual→monolingual and multilingual→multilingual transfer scenarios, the latter being transfer to a much smaller set of languages than the original cross-lingual training set. We compare our vocabulary-replacement techniques against the baseline performance of

XLM-R off-the-shelf, as well as LAPT while retaining the original, full-sized vocabulary.

Another manipulation we consider is whether the transformer-specific parameters are frozen during LAPT. This follows from the literature on transferring monolingual models, which proposes freezing the encoder parameters and only training the new embedding matrix to mitigate catastrophic forgetting during transfer learning (Artetxe et al., 2020; de Vries and Nissim, 2021). In our tables, we denote LAPT with trainable transformer layers as LAPT-FULL, and training with the transformer frozen (but trainable embeddings) as LAPT-EMB.

Target Languages We select our target languages for a wide selection of language families, scripts, typological characteristics, and resource availability, while still having standard evaluation sets for comparison. Training data for all languages is obtained from OSCAR v.22.01 (Abadji et al., 2022). For our lowest-resource languages, supplemental data is obtained from monolingual splits of the OPUS translation corpus (Tiedemann and Nygaard, 2004) and the Johns Hopkins University Bible Corpus (McCarthy et al., 2020). More data curation details may be found in Appendix B.1.

Our multilingual→monolingual transfer languages can be found in Table 3.1. In these experiments, the replacement vocabulary and LAPT training are constrained to a single target language. In addition, we include two multilingual→multilingual experiments. In the first, we simply transfer to the set of languages used in our monolingual experiments. Most of these languages are unrelated and cover a variety of scripts and levels of resource-availability. In the second, we transfer to a set of languages belonging to a single language family — Uralic. These languages come from the same ancestor language, and share broad grammatical features, but also use both Cyrillic and Latin scripts. These differing settings are designed to demonstrate whether language relatedness has an effect on the success of multilingual vocabulary-replacement techniques.

Vocabulary Replacement / Re-initialization When replacing model vocabulary, we train new SentencePiece models on a subset of the training data. For targets with less than 1GB of data, we use the entire dataset. For those with more, we use a random subset of

about 250MB. For multilingual models, we sample 5 million lines according to the same distribution as the training data. All new SentencePiece models have a total vocabulary size of 32,770 including special tokens. We then initialize the embedding matrix for each new vocabulary according to one or a combination of the techniques described in § 3.3.[3]

Training All of our experiments use XLM-R as a starting point (base size; Conneau et al., 2020a). We conduct LAPT for 100k training steps, with evaluation checkpoints every 1000 steps. For LAPT-FULL experiments, the transformer blocks are frozen for the first 10k steps, then unfrozen for the last 90k, so that the model does not overfit to initial (possibly poor) embedding initializations. For LAPT-EMB experiments, transformer blocks remain frozen throughout training. The checkpoint obtaining the best MLM loss on a development set is selected for task fine-tuning and evaluation.

For multilingual training, we sample languages according to a multinomial distribution parameterized by $\alpha = 0.2$, following Conneau and Lample (2019), Conneau et al. (2020a), i.a. Languages are sampled sentence-wise rather than batch-wise.

Evaluation We evaluate model quality with POS-tagging and NER tasks. For each task and each language, the trained model is fine-tuned on task training data until evaluation set convergence or the maximum number of epochs is reached, across four random seeds. POS performance is evaluated on Universal Dependencies (UD) treebanks (de Marneffe et al., 2021), and NER is measured on the WikiAnn benchmark (Pan et al., 2017).

3.5 Results

The results for monolingual adaptation can be found in Tables 3.1-3.2 and general multilingual adaptation in Tables 3.3-3.4. Because the results for multilingual adaptation to the Uralic family mostly echo overall trends, we provide these results in Appendix B.3.[4] In order

[3]The auxiliary FastText model for FOCUS initialization is trained on the same set as the vocabulary

[4]While training on related languages may be beneficial for low-resource Uralic languages like Erzya, family-based training vs. general multilingual training does not seem to alter the relative ranking of embedding initialization techniques, which is our primary research interest

LAPT	REINIT	Armenian	Basque	Erzya	Estonian	Hebrew	Russian	North Sami	Telugu	Avg
*	*	93.4 ± 2.2	95.1 ± 0.7	56.3 ± 5.3	95.6 ± 0.1	97.5 ± 0.1	98.6 ± 0.1	71.2 ± 1.8	83.8 ± 0.1	86.4
FULL	*	-	-	85.1 ± 1.8	-	97.5 ± 0.1	-	-	91.4 ± 4.3	-
FULL	FOCUS+IDENT	**92.3 ± 1.9**	**96.0 ± 0.6**	76.1 ± 2.0	**95.1 ± 0.3**	**97.2 ± 0.1**	**98.4 ± 0.1**	**92.1 ± 0.8**	**86.9 ± 3.5**	**91.7**
FULL	SCRIPT+POSN+IDENT	**93.1 ± 1.7**	93.8 ± 0.5	**79.0 ± 0.7**	94.0 ± 0.2	96.7 ± 0.1	98.2 ± 0.04	86.9 ± 0.7	**88.5 ± 3.2**	91.3
FULL	SCRIPT+IDENT	91.7 ± 1.9	93.6 ± 0.3	70.8 ± 12.8	94.0 ± 0.1	96.7 ± 0.1	98.1 ± 0.1	83.4 ± 1.3	**87.1 ± 3.4**	89.4
FULL	SCRIPT+POSN	90.9 ± 2.0	92.1 ± 0.7	74.6 ± 2.2	90.4 ± 0.6	95.4 ± 0.1	97.2 ± 0.02	78.7 ± 0.5	**87.5 ± 1.4**	88.3
FULL	SCRIPT	89.6 ± 1.5	90.9 ± 0.2	71.5 ± 2.1	89.4 ± 0.9	95.0 ± 0.05	96.9 ± 0.03	77.9 ± 0.2	84.0 ± 1.5	86.9
FULL	IDENT	81.6 ± 0.4	83.6 ± 0.6	59.1 ± 3.1	86.4 ± 0.4	91.1 ± 0.1	96.2 ± 0.04	70.7 ± 0.5	78.0 ± 2.5	80.9
FULL	RANDOM	67.4 ± 2.0	72.7 ± 0.6	53.3 ± 2.8	72.0 ± 0.1	81.0 ± 0.6	86.5 ± 0.6	64.7 ± 0.9	76.4 ± 1.0	72.4
EMB	FOCUS+IDENT	**92.3 ± 1.7**	**95.1 ± 0.6**	48.6 ± 0.1	**94.5 ± 0.05**	**96.9 ± 0.3**	**98.3 ± 0.04**	**73.6 ± 1.6**	**86.2 ± 3.8**	**84.8**
EMB	SCRIPT+POSN+IDENT	87.6 ± 1.3	88.2 ± 0.7	**55.6 ± 4.8**	89.6 ± 0.1	95.3 ± 0.1	97.1 ± 0.05	69.8 ± 1.4	81.8 ± 1.2	82.5
EMB	SCRIPT+IDENT	87.7 ± 1.8	87.9 ± 0.4	**53.8 ± 5.4**	89.2 ± 0.5	95.2 ± 0.1	97.0 ± 0.1	68.6 ± 1.8	82.0 ± 1.3	82.0
EMB	SCRIPT+POSN	56.5 ± 7.6	61.3 ± 12.0	48.7 ± 0.1	71.4 ± 1.4	82.5 ± 0.3	92.1 ± 0.4	59.8 ± 1.5	70.1 ± 7.4	69.4
EMB	SCRIPT	47.6 ± 6.4	59.6 ± 8.1	48.6 ± 0.1	65.7 ± 5.2	80.4 ± 2.2	89.7 ± 1.0	55.5 ± 5.0	73.4 ± 5.5	67.6
EMB	IDENT	80.3 ± 1.1	80.1 ± 0.6	47.9 ± 1.5	82.5 ± 1.8	88.7 ± 0.2	95.2 ± 0.4	60.6 ± 1.2	76.6 ± 1.4	75.9
EMB	RANDOM	47.6 ± 1.8	55.2 ± 2.8	46.3 ± 0.2	63.5 ± 1.8	67.6 ± 2.5	80.2 ± 0.6	44.7 ± 4.0	56.7 ± 6.7	59.2

Table 3.1: Monolingual Language-Adaptive Pre-Training (LAPT): POS tagging accuracy after fine-tuning. * indicates XLM-R off-the-shelf. Within each division, best result and results within 1 standard deviation are bolded; overall best result indicated with added underline. Best result determined by *mean* - *stdev*. LAPT with full XLM-R vocab only conducted for three languages due to prohibitive computational cost

to adhere to our overall computational budget, we only conduct full-vocabulary LAPT experiments for three languages in the monolingual setting.[5]

We first note that across re-initialization methods, LAPT-FULL always outperforms LAPT-EMB. I.e. training with trainable transformer layers outperforms training with frozen ones, despite the risk of catastrophic forgetting with the former. This trend persists across monolingual and multilingual experiments. For example, REINIT-FOCUS+IDENT shows a 6.9 average POS accuracy drop between LAPT-FULL and LAPT-EMB (Table 3.1).

Second, although FOCUS is the best performing re-initialization method when averaged across languages, for individual languages, it does not perform significantly differently than script-based methods. For instance, Armenian and Telugu POS tagging with script-based

[5]We select Erzya, Telugu, and Hebrew for these full-size experiments, spanning very-low, low, and medium resource-availability levels

LAPT	REINIT	Armenian	Basque	Erzya	Estonian	Hebrew	Russian	Telugu	Avg
*	*	94.1 ± 0.1	94.3 ± 0.1	89.5 ± 0.6	93.3 ± 0.2	85.9 ± 0.1	90.9 ± 0.2	85.4 ± 0.5	90.5
FULL	*	-	-	91.8 ± 0.5	-	86.9 ± 0.1	-	86.6 ± 1.9	-
FULL	FOCUS+IDENT	**95.1 ± 0.9**	**94.9 ± 0.4**	**89.9 ± 0.8**	**92.6 ± 0.2**	**86.2 ± 0.3**	**90.6 ± 0.1**	**87.7 ± 0.5**	**91.0**
FULL	SCRIPT+POSN+IDENT	93.9 ± 0.1	94.3 ± 0.2	**90.2 ± 0.7**	92.0 ± 0.3	83.2 ± 0.4	89.8 ± 0.2	83.5 ± 1.8	89.6
FULL	SCRIPT+IDENT	93.8 ± 0.3	94.3 ± 0.1	**89.8 ± 0.2**	89.3 ± 0.2	83.4 ± 0.3	89.4 ± 0.2	84.0 ± 0.5	89.5
FULL	SCRIPT+POSN	92.0 ± 0.6	92.1 ± 0.04	89.1 ± 0.5	88.3 ± 0.4	78.7 ± 0.1	86.5 ± 0.1	81.0 ± 0.9	86.8
FULL	SCRIPT	91.4 ± 0.4	91.1 ± 0.1	87.7 ± 0.5	87.5 ± 0.2	78.5 ± 0.2	85.7 ± 0.1	79.6 ± 1.1	85.9
FULL	IDENT	86.2 ± 0.4	90.7 ± 0.2	79.0 ± 0.6	89.3 ± 0.2	72.0 ± 0.4	86.7 ± 0.1	69.3 ± 0.4	81.9
FULL	RANDOM	74.1 ± 1.4	81.5 ± 0.3	72.6 ± 3.3	45.8 ± 27.2	54.4 ± 0.9	70.3 ± 0.7	47.2 ± 8.2	63.7
EMB	FOCUS+IDENT	**93.5 ± 0.5**	**94.2 ± 0.2**	81.7 ± 2.2	**92.0 ± 0.2**	**84.9 ± 0.1**	**90.3 ± 0.1**	**86.1 ± 0.3**	**89.0**
EMB	SCRIPT+POSN+IDENT	91.5 ± 0.2	92.3 ± 0.1	**87.2 ± 0.3**	89.8 ± 0.2	79.1 ± 0.2	88.9 ± 0.1	74.1 ± 1.2	86.1
EMB	SCRIPT+IDENT	90.9 ± 0.3	92.0 ± 0.3	86.1 ± 1.0	89.6 ± 0.3	78.7 ± 0.3	88.6 ± 0.1	79.1 ± 0.5	86.4
EMB	SCRIPT+POSN	86.5 ± 0.4	87.3 ± 0.3	84.1 ± 1.2	81.8 ± 0.8	71.0 ± 0.9	81.0 ± 0.2	64.3 ± 1.9	79.4
EMB	SCRIPT	83.9 ± 0.4	73.0 ± 0.8	84.0 ± 1.2	79.5 ± 0.9	67.8 ± 0.6	77.4 ± 0.2	56.8 ± 3.2	74.6
EMB	IDENT	80.9 ± 0.8	87.9 ± 0.4	61.8 ± 3.8	85.3 ± 0.3	64.8 ± 1.4	84.8 ± 0.4	54.9 ± 1.5	74.3
EMB	RANDOM	59.6 ± 2.5	0.0 ± 0.0	51.8 ± 2.7	0.0 ± 0.0	17.1 ± 17.2	47.5 ± 6.9	22.4 ± 5.5	28.3

Table 3.2: Monolingual LAPT: entity-wise NER F1 score after fine-tuning. A score of 0.0 results from the model learning to output only class `O` (not a named entity) which is the majority class. Sami does not have enough NER data for fine-tuning

initialization performs on-par with or better than FOCUS (Tables 3.1, 3.3).[6] In the case of the very low-resource language Erzya, script-based methods mostly outperform FOCUS.[7]

Third, for the languages with the largest amount of data in XLM-R (Estonian, Hebrew, and Russian), the off-the-shelf performance of XLM-R (top row) is slightly better than any re-initialization method. This is not unexpected, since we can expect the highest-resource languages in XLM-R to receive adequate vocabulary coverage, and their embeddings are likely the most robustly trained.

Finally, LAPT with the full, original XLM-R vocabulary, results in marginally better performance than other techniques. On one hand, this might be surprising given the

[6]Overall performance/ranking of SCRIPT+POSN+IDENT vs. SCRIPT+IDENT remains uncertain. For LAPT-FULL averaged across languages, the former performs better in 2/3 POS settings, but only 1/3 NER settings

[7]However, script-based methods show significant variation on Erzya POS after multilingual training (Table 3.3)

LAPT	REINIT	Armenian	Basque	Erzya	Estonian	Hebrew	Russian	North Sami	Telugu	Avg
*	*	93.4 ± 2.2	95.1 ± 0.7	56.3 ± 5.3	95.6 ± 0.1	97.5 ± 0.1	98.6 ± 0.1	71.2 ± 1.8	83.8 ± 0.1	86.4
FULL	*	91.3 ± 0.1	95.9 ± 0.6	71.7 ± 5.3	95.5 ± 0.2	97.4 ± 0.2	98.6 ± 0.04	80.6 ± 1.4	89.7 ± 3.6	90.1
FULL	FOCUS+IDENT	91.0 ± 0.1	**95.8 ± 0.1**	**72.5 ± 1.3**	**95.5 ± 0.2**	**97.1 ± 0.1**	**98.4 ± 0.03**	**80.4 ± 1.2**	**89.4 ± 3.2**	**90.0**
FULL	SCRIPT+POSN+IDENT	**92.9 ± 2.1**	95.0 ± 0.6	**63.6 ± 9.8**	94.8 ± 0.3	97.0 ± 0.1	**98.4 ± 0.04**	**80.4 ± 1.1**	**89.6 ± 2.6**	89.0
FULL	SCRIPT+IDENT	**93.8 ± 1.8**	95.3 ± 0.03	**66.1 ± 10.2**	94.7 ± 0.2	**97.1 ± 0.1**	**98.4 ± 0.03**	**80.1 ± 1.2**	**91.7 ± 0.8**	89.7
FULL	SCRIPT+POSN	85.3 ± 3.5	87.9 ± 3.5	70.5 ± 1.5	89.0 ± 0.8	93.7 ± 0.6	97.2 ± 0.01	72.8 ± 2.1	81.6 ± 0.4	84.7
FULL	SCRIPT	83.3 ± 1.9	85.8 ± 2.7	66.6 ± 1.9	85.4 ± 1.7	90.5 ± 0.8	96.8 ± 0.03	68.6 ± 1.1	81.0 ± 0.3	82.2
FULL	IDENT	**93.2 ± 0.7**	93.0 ± 0.5	58.1 ± 0.9	93.6 ± 0.2	96.6 ± 0.1	98.3 ± 0.03	71.5 ± 1.2	89.0 ± 4.1	86.7
FULL	RANDOM	64.5 ± 2.9	67.4 ± 0.4	50.0 ± 4.6	71.9 ± 0.3	80.0 ± 0.8	84.6 ± 0.9	62.7 ± 0.5	75.0 ± 6.2	70.2
EMB	FOCUS+IDENT	**93.1 ± 2.2**	**95.2 ± 0.7**	**63.7 ± 2.0**	**94.7 ± 0.1**	**97.1 ± 0.04**	**98.5 ± 0.03**	71.2 ± 2.1	**87.5 ± 2.9**	**86.8**
EMB	SCRIPT+POSN+IDENT	**91.3 ± 1.6**	93.5 ± 0.6	57.2 ± 7.0	93.5 ± 0.1	96.7 ± 0.03	98.3 ± 0.1	**74.5 ± 1.1**	**85.6 ± 2.9**	85.6
EMB	SCRIPT+IDENT	**92.2 ± 2.0**	93.2 ± 0.7	58.5 ± 6.9	93.3 ± 0.1	96.9 ± 0.1	98.3 ± 0.02	**72.0 ± 3.0**	**86.5 ± 2.4**	85.5
EMB	SCRIPT+POSN	61.5 ± 1.9	76.0 ± 1.3	51.9 ± 3.1	75.7 ± 0.2	87.2 ± 1.2	95.3 ± 0.3	65.3 ± 0.2	77.3 ± 0.3	75.5
EMB	SCRIPT	44.7 ± 0.0	71.0 ± 1.0	48.5 ± 0.2	73.5 ± 2.2	83.6 ± 0.3	93.5 ± 0.5	63.8 ± 1.4	77.7 ± 0.5	73.1
EMB	IDENT	89.4 ± 0.8	90.5 ± 0.6	49.3 ± 4.6	91.8 ± 0.5	96.2 ± 0.1	98.1 ± 0.1	65.6 ± 1.1	84.0 ± 1.7	82.2
EMB	RANDOM	48.7 ± 2.4	61.2 ± 5.6	46.0 ± 0.3	66.3 ± 3.9	73.7 ± 3.4	85.1 ± 1.2	44.7 ± 4.6	67.5 ± 5.0	63.5

Table 3.3: Multilingual LAPT: POS tagging accuracy after fine-tuning

inefficiency with which cross-lingual vocabularies often tokenize low-resource languages (Ács, 2019). On the other hand, these original pre-trained embeddings are also likely robustly aligned with the transformer encoder, which might contribute to slightly better performance.

Part of the motivation for this work, however, is to investigate *efficient* ways to specialize multilingual models. LAPT with the full XLM-R vocabulary is much more computationally costly than training new vocabulary. Figure 3.5 shows the tradeoff between computation (in FLOPs) and performance gain in our experiments: the (often) small gains in performance we see from fine-tuning with the original vocabulary come at the cost of two to three times more FLOPs during adaptation.

Erzya POS performance provides one exception to the pattern of full-vocab LAPT providing only marginal benefits (85.1 accuracy with the full vocabulary vs. 79.0 with the reduced vocabulary). This seems surprising, given Erzya is not included in XLM-R's pre-training data, and intuitively should benefit the most from a specialized vocabulary. It could be that the reduced vocabulary size of 32k is sub-optimal for this particular target language, and/or that the new vocabulary does not overlap enough with the original (full-size) one to inherit useful Cyrillic-script embeddings. Investigating the dynamics of target vocabulary

LAPT	REINIT	Armenian	Basque	Erzya	Estonian	Hebrew	Russian	Telugu	Avg
*	*	94.1 ± 0.1	94.3 ± 0.1	89.5 ± 0.6	93.3 ± 0.2	85.9 ± 0.1	90.9 ± 0.2	85.4 ± 0.5	90.5
FULL	*	94.0 ± 0.5	94.5 ± 0.2	90.5 ± 0.3	93.7 ± 0.2	86.2 ± 0.1	91.1 ± 0.2	85.9 ± 0.7	90.9
FULL	FOCUS+IDENT	**94.2 ± 0.3**	**94.0 ± 0.2**	**89.6 ± 1.0**	**92.0 ± 0.5**	**85.2 ± 0.1**	**90.0 ± 0.5**	**85.4 ± 0.4**	**90.1**
FULL	SCRIPT+POSN+IDENT	**94.1 ± 0.2**	**94.0 ± 0.1**	88.8 ± 0.9	**92.3 ± 0.1**	85.0 ± 0.2	**90.4 ± 0.1**	84.8 ± 0.4	89.9
FULL	SCRIPT+IDENT	**94.2 ± 0.2**	**94.1 ± 0.2**	**90.1 ± 0.6**	**92.4 ± 0.1**	84.9 ± 0.3	90.3 ± 0.1	84.5 ± 0.2	90.0
FULL	SCRIPT+POSN	91.2 ± 0.5	91.5 ± 0.1	88.9 ± 0.5	88.4 ± 0.4	77.3 ± 0.4	86.3 ± 0.1	76.2 ± 0.4	85.7
FULL	SCRIPT	90.9 ± 0.1	91.3 ± 0.3	86.4 ± 1.9	87.7 ± 0.2	75.8 ± 0.3	85.7 ± 0.1	75.1 ± 0.9	84.7
FULL	IDENT	93.2 ± 0.1	93.4 ± 0.2	80.9 ± 2.4	91.5 ± 0.4	83.5 ± 0.3	89.8 ± 0.1	83.2 ± 0.5	87.9
FULL	RANDOM	69.9 ± 4.4	80.9 ± 0.5	75.2 ± 1.5	70.5 ± 2.1	37.7 ± 21.8	68.6 ± 0.7	42.1 ± 1.6	63.6
EMB	FOCUS+IDENT	**93.9 ± 0.3**	**93.7 ± 0.2**	**89.7 ± 0.4**	**91.9 ± 0.4**	**84.8 ± 0.2**	**89.9 ± 0.3**	**85.2 ± 0.5**	**89.9**
EMB	SCRIPT+POSN+IDENT	**93.7 ± 0.2**	93.5 ± 0.1	87.2 ± 1.0	**91.9 ± 0.2**	84.0 ± 0.2	**89.9 ± 0.2**	84.0 ± 0.5	89.2
EMB	SCRIPT+IDENT	93.3 ± 0.5	93.4 ± 0.2	85.8 ± 1.4	**91.9 ± 0.3**	83.7 ± 0.2	**89.9 ± 0.1**	82.5 ± 1.3	88.7
EMB	SCRIPT+POSN	87.5 ± 0.3	88.8 ± 0.3	81.0 ± 3.1	84.8 ± 0.4	72.8 ± 0.1	82.7 ± 0.3	67.1 ± 1.3	80.7
EMB	SCRIPT	85.2 ± 0.3	81.3 ± 7.1	80.0 ± 1.1	84.3 ± 0.3	68.3 ± 0.9	80.6 ± 1.0	59.7 ± 3.5	77.1
EMB	IDENT	91.2 ± 0.3	92.3 ± 0.2	76.7 ± 1.3	90.8 ± 0.3	81.6 ± 0.2	89.3 ± 0.2	78.6 ± 1.8	85.8
EMB	RANDOM	62.8 ± 0.9	74.9 ± 1.6	66.1 ± 1.1	62.7 ± 1.9	23.9 ± 18.2	53.1 ± 4.7	37.7 ± 2.6	54.4

Table 3.4: Multilingual LAPT: entity-wise NER F1 score after fine-tuning

size during vocabulary specialization would be a fruitful direction for future work.

3.6 Discussion

Embedding-only training is inadequate for multilingual model transfer Our experiments show that language transfer methods developed for monolingual models, which freeze the transformer blocks and re-train only the embedding matrix (Artetxe et al., 2020; de Vries and Nissim, 2021), yield poor results when transferring a multilingual model. This work in the monolingual literature not only keeps transformer layers frozen, but initializes new embeddings randomly. This setup (LAPT-EMB, REINIT-RANDOM) performs much worse than the off-the-shelf baseline in all of our experiments.

It is worth noting that Artetxe et al. (2020) do not necessarily suggest that freezing the main model is the *optimal* language transfer method. However, it does demonstrate that for monolingual→monolingual adaptation, embedding-only training is competitive with an off-the-shelf multilingual model. We see no such comparability in our experiments. We

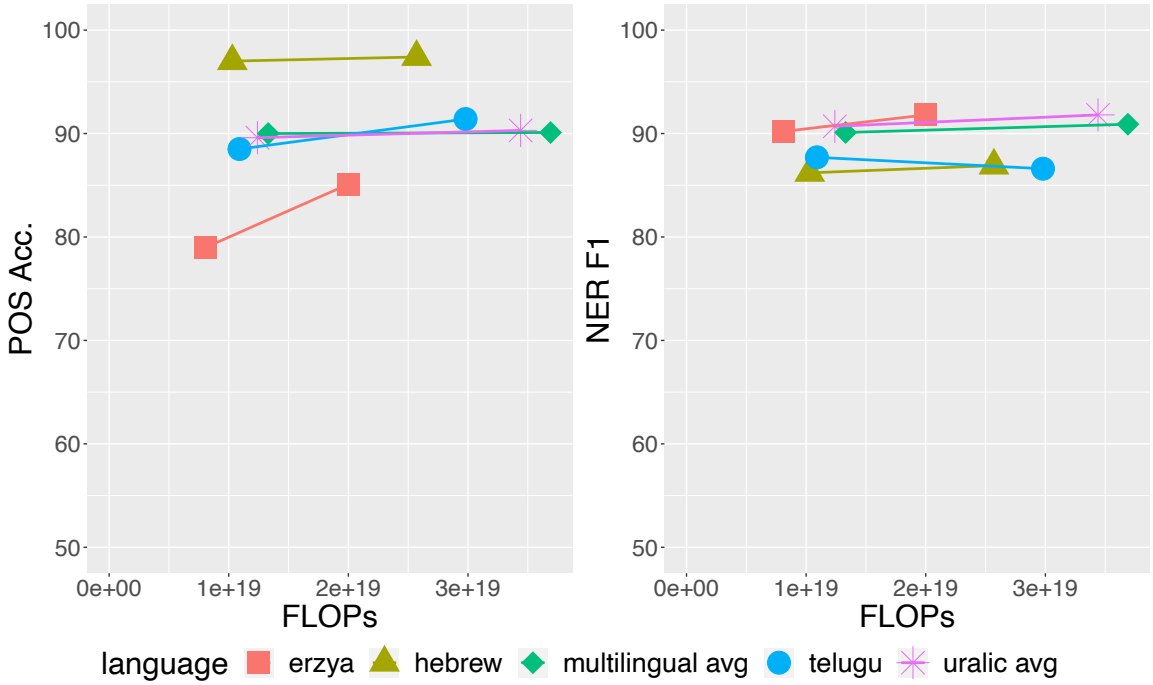


Figure 3.5: Evaluation scores plotted against total floating point operations of LAPT (computational cost). Left point represents cost of LAPT with reduced vocabulary, right point with full vocabulary

believe this is partly caused by the heterogeneity of the XLM-R embeddings, where different languages (or at least scripts) are encoded in different spaces. When new embeddings are randomly and homogeneously initialized, they fail to align with the pre-trained subspaces expected by the frozen transformer.

Vocab replacement efficiently specializes models We demonstrate that for languages inadequately covered by a pre-trained multilingual model, replacing and re-training the cross-lingual model vocabulary with a language-specific one is a computationally efficient way to create a compact model specialized for the target language(s). In our monolingual adaptation experiments, vocabulary replacement performs better than off-the-shelf XLM-R in 5/8 languages for POS tagging and 5/7 languages for NER. Only the high-resource

languages of Estonian, Hebrew, and Russian seem to be adequately covered in XLM-R to outperform our specialization techniques. Language-Adaptive Pre-Training with the full (cross-lingual) XLM-R vocabulary often produces marginally better results overall, but at a much greater computational cost, and without making the model more compact in size. Further training and inference after LAPT will continue to suffer from the memory and compute wasted on unused vocabulary items, which constitute a large percentage of the total model parameters.

Script-distribution initialization rivals semantic similarity methods We introduced several methods for embedding re-initialization in § 3.3, namely using the insight that token embeddings for XLM-R cluster by script and position within a word, then distributing new vocabulary items according to these pre-trained sub-distributions. We compare this to the FOCUS re-initialization method, which initializes new embeddings as a weighted combination of existing ones according to similarity scores from an auxiliary model.

Averaged across languages, FOCUS yields the best performance in downstream tasks by a slight margin. Within languages, it often overlaps significantly with the performance of our script-distribution methods. For very low-resource languages like Erzya, script-based methods even show a slight advantage. This seems to show that, at least in combination with LAPT, the majority of the benefit in re-initialization can be achieved by a method that takes the structure of the pre-trained embedding distribution into account, whether or not it uses advanced methods to precisely initialize the representations of new vocabulary items.

We do note that the advantage of FOCUS is more clear-cut when LAPT is conducted with transformer blocks frozen. This lends credence to the idea that FOCUS more precisely mimics the embedding distribution expected by the pre-trained transformer. However, the overall best results come when the transformer blocks are unfrozen/trainable.

Fully random initialization performs poorly Finally, our experiments demonstrate that fully random re-initialization of embeddings during vocabulary replacement leads to overall poor performance. Across LAPT-FULL experiments, random initialization performs an average of 19.4 points worse than the next-best re-initialization method, and 24.7 points

worse than the off-the-shelf baseline. The poor performance of random initialization has been noted in other works such as Dobler and de Melo (2023), but we emphasize that even incredibly simple methods such as REINIT-IDENT and REINIT-SCRIPT work far better than the random baseline.

3.7 Limitations

One limitation of our work is the relatively narrow set of evaluation tasks available for our languages of interest. The model-adaptation techniques we compare here are most applicable to low- and medium-resource languages that are not optimally covered by pre-existing multilingual models. For most of these languages, the only standard evaluation datasets that exist are for relatively low-level tasks like Part of Speech tagging and Named Entity Recognition. Evaluation of embedding-reinitialization techniques could be improved in future work if datasets for higher-level tasks like Natural Language Inference, question answering, and paraphrase detection were curated for these under-resourced languages.

We also make several simplifying choices to maintain a feasible scope for our work. First, we conduct model adaptation from only a single base model: XLM-R. A valuable addition in future work would be to determine whether the trends we observe here generalize to other model types (i.e. causal and seq2seq language models) and to larger model scales. Secondly, we consider only one size for newly-initialized target vocabularies (32k). Because effective per-language vocabulary allocation has been shown to be an important factor in multilingual modeling (Conneau et al., 2020a, i.a.), investigating the dynamics of target vocabulary size during vocabulary re-initialization will be important for future work on this topic.

3.8 Conclusion

This work presents a systematic comparison of methods to specialize the subword vocabularies and embeddings of multilingual models for new languages. We propose simple methods for re-initializing embeddings, motivated by a qualitative exploration of the XLM-R embedding space. Our experiments show that (1) updating the encoder layers during LAPT is crucial for downstream performance, (2) vocabulary replacement provides a computationally-efficient method to improve task performance in low-resource languages, and (3) our re-initialization

techniques employing script-wise sub-distributions perform on par with more involved similarity-based methods. We hope these findings can be built upon in future work on multilingual model specialization, with the goal of providing the best performance for under-resourced languages while also making language modeling more accessible through more manageable compute cost and model sizes.

Relation to Remaining Work

Chapter 4 will directly follow up on the results of this chapter by applying vocabulary specialization to the process of adapting a cross-lingual model to a specific language family. It will also address the outstanding question of how much the size of the adapted vocabulary affects downstream model performance. Instead of further comparing techniques for embedding re-initialization, we settle on the FOCUS algorithm as the overall best performing, and turn our attention to other adaptation dynamics such as multilingual vs. monolingual adaptation, number of training steps, and language sampling proportions during training. Similar to the current chapter, Chapter 4 will contribute both to optimizing model performance in a set of target languages, and to reducing model size by cutting unnecessary parameters.

Chapter 4

TARGETED MULTILINGUAL ADAPTATION FOR LOW-RESOURCE LANGUAGE FAMILIES

Overview

This chapter presents a systematic analysis of the best methodology for adapting a pre-trained cross-lingual model to a language family. Family-wise adaptation is one instantiation of our broader principle of targeted multilingual training — providing a middle-ground between the largely intractable option of training a separate model for each individual language, and a "massively multilingual" setting, which is known to lead to poor performance in under-resourced languages. Our models — adapted to the Uralic family via unsupervised training and vocabulary specialization — significantly outperform both monolingual and massively multilingual baselines. While vocabulary specialization was explored in Chapter 3, we here address the unanswered question of how much the *size* of the adapted vocabulary matters for downstream performance. A statistical analysis of adaptation parameters reveals that though both the number of adaptation steps and specialized vocabulary size positively contribute to model performance, training for longer is significantly more effective than choosing a larger vocabulary. We also show that low-resource languages can be aggressively up-sampled during adaptation, without significantly hurting performance in the high-resource languages that are down-sampled to compensate.

4.1 Introduction

Pre-trained multilingual language models act as the foundation for most current NLP systems outside of English and a few other very high-resource languages. While most languages of the world are relatively data-scarce in comparison to English, multilingual models take the approach of pooling text data across many languages to train a single model that (in theory) covers all training languages (Devlin, 2019; Conneau and Lample, 2019; Conneau et al.,

2020a; Liu et al., 2020; Scao et al., 2023, i.a.). In practice, however, massively-multilingual models often perform poorly on low-resource languages (Wu and Dredze, 2020).

While multilingual models are susceptible to the so-called "curse of multilinguality" — the observation that overall model performance decreases as more languages are added in pre-training (Conneau et al., 2020a; Wang et al., 2020b) — it is generally accepted that low-resource languages benefit from *some* multilinguality during training, especially when added languages are similar in some way (Conneau et al., 2020a; Ogunremi et al., 2023; Chang et al., 2023). Nonetheless, "massively multilingual" or "cross-lingual" models have remained a central focus of multilingual LLM research (e.g. Üstün et al., 2024).

This paper joins a growing line of research studying *targeted* multilingualism as a more practical approach to building robust models for mid- and low-resource languages (Chang et al., 2023; Ogueji et al., 2021; Ogunremi et al., 2023; Ljubešić et al., 2024). While studies like Ogunremi et al. (2023) take the approach of training from scratch on a linguistically-informed grouping like a language family, we instead seek to determine the best way to *leverage* existing multilingual models, using their parameters as a starting point for specialization to a more moderate set of languages.

In this work, we systematically evaluate the best technique for adapting a pre-trained multilingual model (XLM-R) to a language family. We use the Uralic family as a case study — like many families, it includes a few mid-resource languages (e.g. Hungarian, Finnish) as well endangered and Indigneous languages like Sámi and Erzya, which are extremely data-scarce. Our primary techniques for conducting adaptation are multilingual Language-Adaptive Pre-Training (LAPT; Chau et al., 2020) and vocabulary replacement/specialization (Dobler and de Melo, 2023; Downey et al., 2023, i.a.). Our experiments show that both techniques are necessary for robust adapation to the Uralic family.

Importantly, we demonstrate not only that adaptation to a language family is as effective or better than training individual models, but also that it is more efficient than monolingual adaptation. We also statistically analyze important factors in multilingual adaptation in order to recommend *best practices* for adapting models to new language families, as measured by down-stream task performance. In particular, we use a regression analysis to assess the impact of LAPT steps, adapted vocabulary size, and language sampling alpha on model

performance. Notable results include the fact that specialized vocabularies as small at 16k tokens outperform the cross-lingual XLM-R vocabulary (with 250k tokens), and low-resource languages can be aggressively up-sampled during training without significant degradation of high-resource performance (see § 4.4,4.5 for more details).

Our contributions are as follows: 1) We train models adapted for the Uralic family that significantly outperform monolingual and multilingual baselines for almost all languages. 2) We conduct a large-scale statistical analysis of important parameters for multilingual adaptation to test their relative effects on downstream task performance. 3) We make best-practice recommendations for adapting cross-lingual models to targeted groupings like language families. 4) We provide an error analysis for Skolt Sámi, which is consistently difficult to model, and discuss the implications and challenges of these results for future work. 5) We make all of our adaptation code, configurations, analysis results, and best-performing Uralic model(s) publicly available at `https://github.com/CLMBRs/targeted-xlms`.

4.2 Related Work

Pre-trained model adaptation Extensive work has proposed re-using and modifying pre-trained models for new settings in order to retain existing model knowledge and reduce pre-training costs. Gururangan et al. (2020) show that continued training on domain-specific data effectively adapts pre-trained models to new domains in both high- and low-resource settings. This approach is also used to adapt models to new languages (i.e. Language-Adaptive Pre-Training / LAPT; Chau et al., 2020).

Other approaches involve training new, language-specific adapter layers to augment a frozen monolingual Artetxe et al. (2020) or multilingual encoder Pfeiffer et al. (2020); Üstün et al. (2020); Faisal and Anastasopoulos (2022). A comparison of these cross-lingual adaptation approaches Ebrahimi and Kann (2021) found that continued pre-training often outperforms more complex setups, even in low-resource settings.

Ács et al. (2021) investigate the transferability of monolingual BERT models for Uralic languages specifically. They find that vocabulary overlap and coverage is extremely important for transfer success, and also that the importance of language-relatedness is questionable, since English and Russian BERT transfer well to Uralic languages written in Latin and

Cyrillic script, respectively.

Model vocabulary and script A major limitation to adapting pre-trained models to new languages is the subword vocabulary, which often fails to cover unseen scripts Pfeiffer et al. (2021) or tokenizes target text inefficiently Ács (2019); Ahia et al. (2023). Muller et al. (2021) demonstrate that script is another important factor in predicting transfer success: pre-trained coverage of closely-related languages improves transfer, but only if the target language is written in the same script as its pre-trained relative.

A range of adaptation techniques have been proposed to overcome this tokenization issue, such as extending the vocabulary with new tokens (Chau et al., 2020; Wang et al., 2020a; Liang et al., 2023) or completely replacing and re-training the vocabulary and embedding matrix from a random initialization (Artetxe et al., 2020; de Vries and Nissim, 2021). Other work reuses information in pre-trained embeddings rather than initializing new ones at random. This may include scaling up smaller embedding spaces from models trained on the target language de Vries and Nissim (2021); Ostendorff and Rehm (2023) or copying embeddings from the original vocabulary where there is exact vocabulary overlap Pfeiffer et al. (2021).

In this study, we follow a line of recent work that re-initializes vocabulary and embeddings based on the structure of the embedding space for the original model (Minixhofer et al., 2022; Ostendorff and Rehm, 2023, i.a.). Dobler and de Melo (2023) introduce the FOCUS algorithm, which like Pfeiffer et al. (2021) carries over original embeddings where there is an exact match with the new vocabulary. For new tokens however, it initializes embeddings as a linear combination of the old embeddings for the most semantically similar tokens, as computed by an auxiliary embedding model. As an alternative, Downey et al. (2023) propose three simple heuristics for initializing a new embedding matrix, one being the familiar strategy of carrying over the embeddings of overlapping tokens, and the others involving initializing new tokens based on script-wise distributions in the original space. They compare these methods to the FOCUS algorithm and find the latter has only a small advantage over the heuristic-based techniques.

Targeted multilingualism A recent line of work has proposed models trained with *targeted* or *linguistically-informed* multilingualism, as opposed to the "massively-multilingual" approach covering as many languages as feasible (e.g. Conneau et al., 2020a; Scao et al., 2023). Notably, Chang et al. (2023) show that while massively-multilingual models hurt individual language performance, low-resource languages in particular benefit from *limited* multilinguality, especially when the added languages are syntactically similar (e.g. similar word order).

Examples of targeted multilingual approaches include Ogueji et al. (2021), who train a multilingual model from scratch on 11 African languages and show performance that is as good or better than XLM-R. Ogunremi et al. (2023) refine this approach by showing that multilingual training on languages from individual African language families is more data-efficient than using a mixture of unrelated African languages. Snæbjarnarson et al. (2023) also show success for the low-resource language Faroese by training a multilingual model on its close Germanic relatives.

Other work investigates using multilingual training with related languages as an *adaptation* process, starting from a pre-trained cross-lingual model rather than training from scratch. Alabi et al. (2022) adapt XLM-R to the 17 highest-resource African languages via LAPT, while also removing XLM-R vocabulary items that are unused for the target languages. Ljubešić et al. (2024) use LAPT to adapt XLM-R to the very closely related Slavic languages of Bosnian, Croatian, Montenegrin, and Serbian. Senel et al. (2024) adapt XLM-R separately to five low-resource Turkic languages, showing that including the high-resource Turkish language during training improves this adaptation.

The present work systematically analyzes which factors are responsible for the success of targeted multilingual adaptation. We focus on the model adaptation paradigm since cross-lingual models learn useful language-general patterns that can be leveraged for a "warm-start" to training (Conneau et al., 2020b). Unlike Ljubešić et al. (2024); Senel et al. (2024), we specialize model vocabulary for the target language(s), since cross-lingual tokenizers typically perform poorly for low-resource languages (Rust et al., 2021). We follow Dobler and de Melo (2023) and Downey et al. (2023) in using a vocabulary specialization technique that leverages the structure of the original model embedding space, while creating a new vocabulary that

is directly optimized for the target languages, in contrast to Alabi et al. (2022), which simply uses a subset of the original model vocabulary. Finally, we follow Ogunremi et al. (2023) in conducting adaptation for a language family, while keeping in mind the observation from Senel et al. (2024) that including a high-resource language during adaptation can be advantageous. This comes naturally with our chosen testbed of the Uralic family, which contains both high- and low-resource languages.

4.3 Experiments

Our experiments are designed to assess the best method for adapting a pre-trained cross-lingual model to a specific language family (in our case, Uralic). We are especially interested in identifying conditions that produce the best model(s) for low-resource family members. Our primary approach employs Language-Adaptive Pre-Training (LAPT, Chau et al., 2020) on a dataset of Uralic languages, as well as vocabulary specialization (Downey et al., 2023, i.a.). Adapted models are compared to both multilingual and monolingual baselines.

Within our multilingual experiments, we search a range of important hyper-parameters and explicitly model their influence on downstream performance using a linear mixed-effects regression. Namely, we test the effect of number of LAPT steps, size of the language-specialized vocabulary, and the α parameter controlling multinomial language sampling distribution during LAPT (Conneau and Lample, 2019; Conneau et al., 2020a).

Languages The first step of our adaptation process is to obtain raw-text LAPT data for as many Uralic languages as possible. For the high-resource languages (Estonian, Finnish, Hungarian, and Russian), we obtain all training data from the multilingual OSCAR corpus v.22.01 (Abadji et al., 2022). This corpus also contains a small amount of raw text for the low-resource languages Komi (`koi`) and Mari (`mhr/mrj`). We further source low-resource language data from monolingual splits of the OPUS translation corpus (Tiedemann and Nygaard, 2004) and the Johns Hopkins University Bible Corpus (McCarthy et al., 2020).

An inventory of LAPT text data is found in Table 4.1. This represents the total amount of data after combining all corpora for each language. We cover 6/8 Uralic branches, lacking only Ob-Ugric and Samoyedic (Austerlitz, 2008). The resource gap between the high-

Language	Code	Branch	Script	XLM-R Data (GB)	LAPT Data (GB)	LAPT Data (lines)	Sources
Russian	ru	n/a	Cyrillic	278.0	9.1	32.7×10^6	O
Hungarian	hu	Hungarian	Latin	58.4	12.8	64.8×10^6	O
Finnish	fi	Finnic	Latin	54.3	9.3	50.2×10^6	O
Estonian	et	Finnic	Latin	6.1	2.8	15.8×10^6	O
Komi	koi	Permic	Cyrillic	0	6.8×10^{-3}	48.5×10^3	OPJ
Mari	mhr/mrj	Mari	Cyrillic	0	6.5×10^{-3}	25.3×10^3	OJ
Erzya	myv	Mordvinic	Cyrillic	0	6.0×10^{-3}	32.6×10^3	PJ
Veps	vep	Finnic	Latin	0	5.3×10^{-3}	35.7×10^3	P
Udmurt	udm	Permic	Cyrillic	0	4.3×10^{-3}	28.1×10^3	PJ
Sámi	se/sme	Sámi	Latin	0	3.9×10^{-3}	34.5×10^3	PJ
Karelian	krl	Finnic	Latin	0	2.4×10^{-3}	17.4×10^3	PJ
Moksha	mdf	Mordvinic	Cyrillic	0	1.2×10^{-3}	9.3×10^3	P
Livonian	liv	Finnic	Latin	0	0.5×10^{-3}	14.2×10^3	P
Votic	vot	Finnic	Latin	0	$< 0.1 \times 10^{-3}$	474	P
Ingrian	izh	Finnic	Latin	0	$< 0.1 \times 10^{-3}$	21	P

Table 4.1: Listing of available training data by language (after cleaning, de-duplicating, and reserving 10% for eval and test sets). XLM-R data is the amount of data used to pre-train that model. LAPT data is the amount of data available for adaptive training on Uralic languages in our experiments. Codes for language data sources: O = OSCAR, P = OPUS, J = JHUBC.

and low-resource languages is stark: Estonian (the fourth-highest-resource language) has approximately 1000x more data than the next highest (Komi). These four highest-resource languages were also included in the training data for XLM-R, while the remainder were not. We treat this as the cutoff point between the "high-resource" and "low-resource" Uralic languages for the remainder of this work.

We include Russian as a high-resource language, though it is not Uralic. Many Uralic languages are spoken by ethnic minorities within Russia and the former Soviet Union, and use modified forms of the Russian Cyrillic alphabet. The lack of a high-resource Uralic language written in Cyrillic could be a problem for low-resource language performance, since script overlap has been shown to be a vital ingredient in cross-lingual transfer (Muller et al.,

2021; Downey et al., 2023). Further, Russian is a major source of loan-words for Uralic languages, as well as an official language throughout Russian territory (Austerlitz, 2008).

During our experiments, we sample languages according to a multinomial distribution parameterized by the hyper-parameter α (Conneau and Lample, 2019; Conneau et al., 2020a, i.a.; see Figure 4.1). Languages are sampled sentence-wise rather than batch-wise, meaning multiple languages can be sampled in each batch.

Vocabulary replacement To specialize the model's vocabulary for target languages, we first train a new SentencePiece model (Kudo and Richardson, 2018) on 5 million lines sampled from the training set.[1] For simplicity, we train multilingual tokenizers with a consistent sampling parameter of $\alpha = 0.2$.[2] Once a new vocabulary is formed, we re-initialize the model's embedding matrix using the FOCUS algorithm introduced by Dobler and de Melo (2023). We test the effect of vocabulary size by training specialized vocabularies with 16k, 32k, and 64k tokens.[3]

Training All experiments use XLM-R base as a starting point (Conneau et al., 2020a). We conduct LAPT on the multilingual Uralic dataset for 100k, 200k, or 400k steps. Following Downey et al. (2023), for experiments with vocabulary specialization, the transformer blocks are frozen for the first 10k steps, then unfrozen for the remainder, to prevent model overfitting on the initial (possibly poor) embedding initializations. The checkpoint with the best MLM loss on a development set is selected for task fine-tuning and evaluation.

For our shortest experiments (100k steps) we test four values of α: $\{0.1, 0.2, 0.3, 0.4\}$. For longer experiments, we test only the two most promising values: $\{0.1, 0.2\}$. Because the data ratio between our high and low-resource languages is so extreme (Table 4.1), we cap the four high-resource languages at approximately 2 GB of text each.[4] Because several languages of

[1]When adapting to single languages with < 5 million lines, the vocabulary is trained on the entire training set.

[2]Pilot experiments suggest the choice of α during vocabulary initialization is not as important as the value picked during multilingual training.

[3]Throughout this paper, 16k, 32k, and 64k are shorthand for 2^{14}, 2^{15}, and 2^{16} respectively.

[4]This is in addition to alpha sampling, reflected in Figure 4.1.

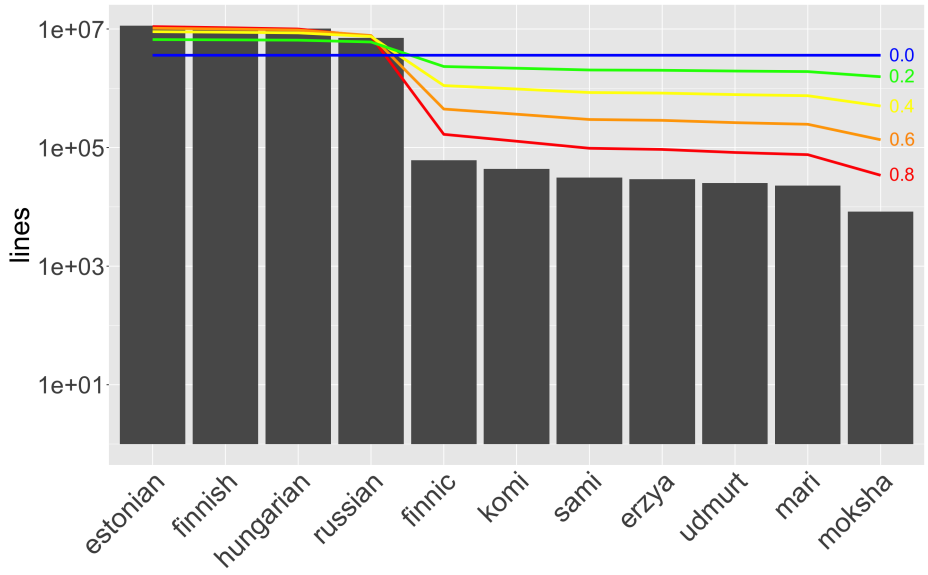


Figure 4.1: Uralic data composition by number of lines, on a log scale. The actual data quantities are shown with bars, while sampling distributions with several values of the α parameter are plotted as lines

the Finnic branch have less than 1 MB of text, we also sample the 5 low-resource Finnic languages as if they are a single language ("Finnic" in Figure 4.1). This is to prevent extreme over-sampling of tiny datasets such as Ingrian.

Task evaluation We evaluate model performance with Part Of Speech (POS) tagging accuracy as well as Unlabeled Attachment Score (UAS), a metric for syntactic dependency parsing. Both of these evaluations are conducted on Universal Dependencies (UD) treebanks de Marneffe et al. (2021).[5] Treebanks are available for all high-resource languages plus Erzya, North Sámi (`sme`), Komi, Karelian, Livvi, Moksha, and Skolt Sámi (`sms`). Models are fine-tuned for each task over four random seeds.

Because the available amount of fine-tuning data varies considerably over languages, we

[5]Currently, UD appears to be the only source for high-quality NLP evaluation data in low-resource Uralic languages.

consider three evaluation settings: *few-shot*, *full-finetune*, and *zero-shot*. In the *few-shot* setting, models are fine-tuned on 512 sampled sentences per language. For *full-finetune*, models are fine-tuned on the entirety of the fine-tuning data for each language (ranging from 896 sentences for Erzya to 32,768 for Russian). We additionally employ the *zero-shot* setting because, with the exception of Erzya and North Sámi, the low-resource languages we consider only have small test sets, with no standard training data. For this setting, we fine-tune the model on the full collection of languages with training sets, and then evaluate directly on the target test set. An inventory of Uralic UD evaluation data can be found in Table C.2 of Appendix C.2, along with more details on our evaluation methodology.

Baselines Our simplest baseline is "off-the-shelf" XLM-R — the pre-trained model from Conneau et al. (2020a) with no modifications. We also test XLM-R adapted with LAPT, but without vocabulary specialization. LAPT alone is a strong baseline. However, as Downey et al. (2023) note, keeping a large "cross-lingual" vocabulary during LAPT incurs considerable extra computational cost compared to training a smaller, specialized vocabulary. Given the observation that cross-lingual tokenizers are inefficient and ineffective for low-resource languages (Ács, 2019; Rust et al., 2021), we hypothesize a specialized vocabulary will show a performance advantage in addition to the reduction in computational cost.

We also compare our multilingual models to baselines adapted to single languages. While multilingualism is known to help low-resource languages to some degree, it is also an open question in what circumstances multilingualism becomes a "curse" (Conneau et al., 2020a; Chang et al., 2023). To make this comparison, we adapt XLM-R with LAPT on individual languages, with a vocab size of 16k per language, and assuming a shared computational "budget" of 400k training steps. The steps are allocated across languages according to the multinomial distribution with $\alpha = 0.1$, similar to the data sampling technique for multilingual training. We thus design this baseline to be roughly comparable to our multilingual model trained with 400k steps, vocab size 16k, and $\alpha = 0.1$.

Task	Type	Erzya	North Sámi	Estonian	Finnish	Hungarian	Russian	Avg
UAS	monolingual	49.7 ± 0.7	42.0 ± 2.2	52.4 ± 1.0	**69.2 ± 2.1**	63.2 ± 3.4	69.1 ± 1.8	57.6
UAS	multilingual	**58.8 ± 2.3**	**51.3 ± 0.5**	**56.9 ± 2.5**	**71.2 ± 2.1**	**69.9 ± 1.2**	**71.7 ± 2.6**	**63.3**
POS	monolingual	62.0 ± 1.3	60.8 ± 2.0	**84.0 ± 0.6**	**79.1 ± 2.3**	85.9 ± 2.2	86.5 ± 1.8	76.4
POS	multilingual	**76.1 ± 3.3**	**73.2 ± 1.2**	77.7 ± 3.9	**79.7 ± 2.6**	**89.3 ± 1.3**	**87.5 ± 0.5**	**80.6**

Table 4.2: Few-shot comparisons with monolingual baselines (both tasks). All models have vocabulary size 16k. Multilingual models are trained for 400k steps with $\alpha = 0.1$. Monolingual models trained for a total of 400k steps "budgeted" across the languages, according to $\alpha = 0.1$, as described in § 4.3.

4.4 Results

We present our results in two main sections. First, we compare our best-performing Uralic-adapted multilingual models to both multilingual and monolingual baselines. We show that our chosen method of layering LAPT and vocabulary specialization on a pre-trained multilingual model largely outperforms alternatives on downstream tasks and is more computationally efficient.

We then analyze the dynamics of important factors during multilingual adaptation such as number of LAPT steps, adapted vocabulary size, and sampling alpha. Our grid search of hyper-parameters for multilingual LAPT yields 72 evaluation data-points per language, per task, per setting.[6] We first visualize and discuss the overall trends observed for each parameter; then, we present a regression analysis of the combined effect of these parameters on task performance.

4.4.1 Baselines

Monolingual baselines Tables 4.2 and 4.3 compare our best-performing, fully-adapted multilingual models to the comparable monolingual baselines described in §4.3. With a few

[6] 3 training lengths × 3 vocabulary sizes × 2 alpha values × 4 random seeds (during fine-tuning) = 72. Only 2 alpha values are tested over all training lengths.

Task	Type	Karelian	Komi	Livvi	Moksha	Skolt Sámi	Avg
UAS	monolingual	61.7 ± 0.4	28.4 ± 4.6	61.1 ± 0.8	40.0 ± 3.1	28.9 ± 2.1	44.0
UAS	multilingual	**65.9 ± 0.3**	**73.8 ± 0.6**	**65.9 ± 0.3**	**70.2 ± 0.2**	**41.4 ± 1.6**	**63.4**
POS	monolingual	84.5 ± 0.1	44.6 ± 3.1	81.6 ± 0.2	49.7 ± 2.0	52.6 ± 0.5	62.6
POS	multilingual	**87.7 ± 0.2**	**80.1 ± 0.3**	**85.0 ± 0.2**	**78.3 ± 0.2**	**55.4 ± 0.3**	**77.3**

Table 4.3: Zero-shot comparisons with monolingual baselines (both tasks) with the same models as Table 4.2. Monolingual models are fine-tuned on the most similar language with a UD training set: Finnish → Karelian, Livvi; Erzya → Komi, Moksha; North Sámi → Skolt Sámi.

exceptions for high-resource languages like Estonian and Finnish, the multilingual models substantially outperform the baselines. This is especially salient for the UAS task (first two rows of each table), the *zero-shot* setting (Table 4.3), and low-resource languages.

Multilingual baselines Tables 4.4 and 4.5 show a comparison of our fully-adapted multilingual models to multilingual baselines for the dependency parsing task. The first row in each represents XLM-R "off-the-shelf" — the original model without LAPT or adjustments to the vocabulary. The second row is the XLM-R adapted with LAPT, but without vocabulary specialization. It retains the large "cross-lingual" vocabulary inherited from XLM-R, which is almost 4x larger than our largest adapted vocabulary (64k tokens).

Table 4.4 shows that in *few-shot* evaluations, our smallest model with vocabulary specialization significantly outperforms the best baseline model without. Creating an adapted vocabulary of 16k tokens results in an average performance gain of 1.6 over the baseline, and increasing to 64k tokens yields an improvement of 4.7 points. We also note that conducting LAPT on XLM-R with its original vocabulary incurs approximately 2-3x more computational cost than training a version with a specialized vocabulary of size 32k (Downey et al., 2023).

In contrast, the *zero-shot* evaluations do not reflect this consistent improvement with

LAPT	Alpha	Vocab	Erzya	North Sámi	Estonian	Finnish	Hungarian	Russian	Avg
0	*	250k (orig)	29.0 ± 2.1	26.2 ± 1.0	37.4 ± 5.4	51.5 ± 3.1	45.3 ± 10.0	47.6 ± 3.5	39.5
400k	0.1	250k (orig)	54.0 ± 0.9	51.0 ± 1.3	54.7 ± 2.3	71.2 ± 1.0	69.1 ± 1.4	70.1 ± 3.4	61.7
400k	0.1	16k	58.8 ± 2.3	51.3 ± 0.5	56.9 ± 2.5	71.2 ± 2.1	69.9 ± 1.2	71.7 ± 2.6	63.3
400k	0.1	32k	56.6 ± 0.8	52.0 ± 0.8	56.7 ± 1.9	72.0 ± 1.8	70.1 ± 0.8	71.9 ± 2.0	63.2
400k	0.1	64k	**61.5 ± 2.8**	**53.8 ± 0.8**	**60.7 ± 0.9**	**73.0 ± 1.0**	**75.2 ± 0.5**	**74.2 ± 2.2**	**66.4**

Table 4.4: Few-shot UAS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocabulary specialization). Second row is XLM-R with original cross-lingual vocabulary, but fine-tuned on Uralic languages with LAPT

LAPT	Alpha	Vocab	Karelian	Komi	Livvi	Moksha	Skolt Sámi	Avg
0	*	250k (orig)	59.0 ± 0.4	41.1 ± 1.4	56.0 ± 0.9	52.7 ± 0.03	44.4 ± 1.4	50.6
400k	0.1	250k (orig)	65.2 ± 0.3	73.9 ± 0.4	63.4 ± 0.4	70.4 ± 0.6	**44.8 ± 1.2**	63.6
400k	0.1	16k	65.9 ± 0.3	73.8 ± 0.6	**65.9 ± 0.2**	70.2 ± 0.2	41.4 ± 1.6	63.4
400k	0.1	32k	**66.4 ± 0.4**	74.9 ± 0.3	65.4 ± 0.7	71.7 ± 0.7	43.3 ± 1.5	**64.3**
400k	0.1	64k	66.0 ± 0.4	**75.0 ± 0.1**	65.6 ± 0.5	**73.3 ± 0.5**	40.8 ± 1.3	64.1

Table 4.5: Zero-shot UAS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocabulary specialization). Second row is XLM-R with original cross-lingual vocabulary, but fine-tuned on Uralic languages with LAPT

increasing adapted vocabulary size (Table 4.5; this is also reflected in our statistical analysis later in this section). 4 of the 5 zero-shot languages still see their best results when modeled with a specialized vocabulary. The exception is Skolt Sámi, which is modeled best by the +LAPT/-vocab-adaptation baseline. However, as we will note several times, our results for Skolt Sámi go against overall trends in our experiments, and we delve into this finding further with an error analysis in §4.5.

For space and clarity, we have focused only on the UAS results in this section. The comparable tables for POS can be found in Appendix C.3. For POS, we observe similar

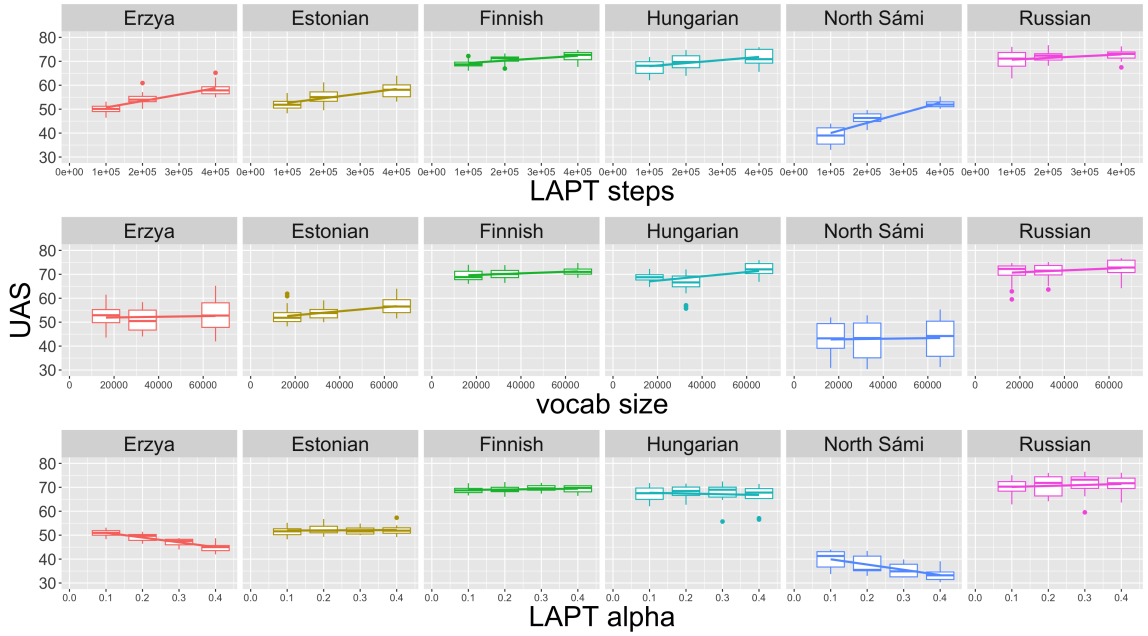


Figure 4.2: Few-shot UAS — effect of hyper-parameters on task performance, by language. Plots for each individual parameter are marginalized across values of the remaining parameters. We test the following values: LAPT steps: {100k, 200k, 400k}, vocabulary size: {16k, 32k, 64k}, LAPT alpha: {0.1, 0.2, 0.3, 0.4}.

trends to UAS, though the LAPT baseline with the original vocabulary is more on par with the specialized vocabulary settings. We hypothesize that this is reflective of POS tagging being an overall simpler task than dependency parsing, since the latter requires more advanced knowledge of linguistic structure. We believe it is telling, therefore, that the advantage of specialized-vocabulary models is clearer in the more complicated UAS task.

4.4.2 Qualitative trends

Figure 4.2 shows visualizations of the per-language effect of each hyper-parameter (marginalized across other parameters) in the *few-shot* setting. These plots show the UAS experiments,

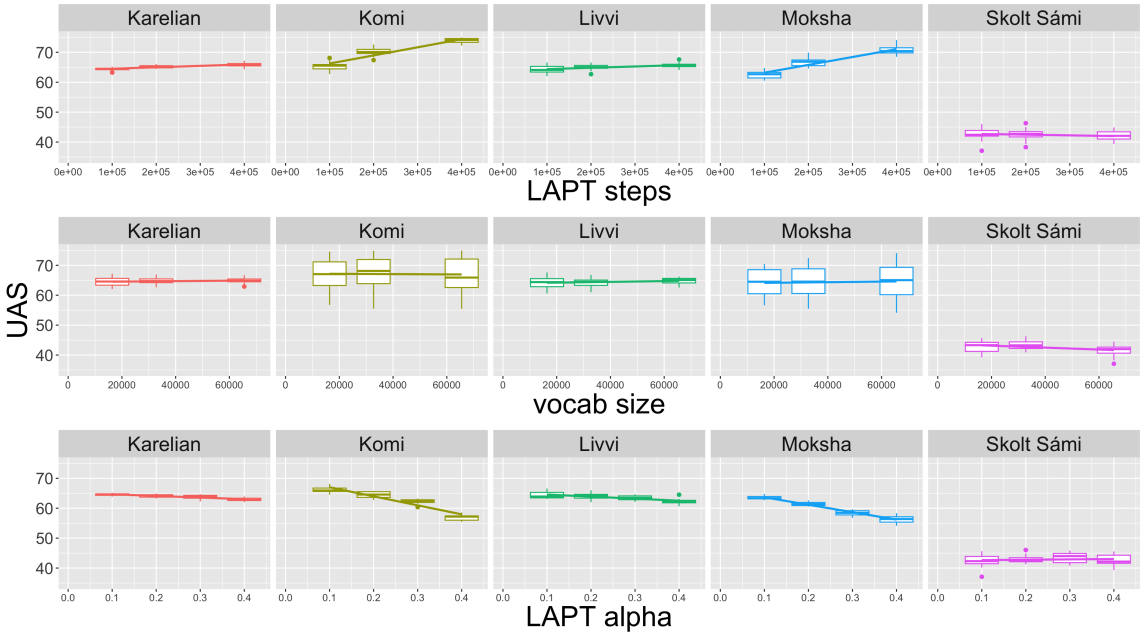


Figure 4.3: Zero-shot UAS — effect of hyper-parameters on task performance, by language. Plots for each individual parameter are marginalized across values of the remaining parameters. We test the following values: LAPT steps: {100k, 200k, 400k}, vocabulary size: {16k, 32k, 64k}, LAPT alpha: {0.1, 0.2, 0.3, 0.4}.

but they reflect overall trends seen in our statistical data analysis across both tasks.[7] First, the number of LAPT (training) steps unsurprisingly has a large effect on performance across languages; this reflects that the adapted model may take a long time to properly converge on new languages. This may be supported by the slope being steeper for languages that are new to XLM-R such as Erzya (myv). Second, adapted vocabulary size seems to have an overall positive effect on performance. However, this effect is not as strong as adding more LAPT steps and not as clear for the low-resource languages Erzya (myv) and North Sámi (sme). Finally, the effect of sampling alpha diverges between high- and low-resource languages, as lower alpha values up-sample low-resource languages and down-sample high-resource ones.

[7] A corresponding visualization for POS can be found in Figure C.1 in the Appendix.

More notable is the fact that the performance gain for low-resource languages at lower alpha values is much greater than the corresponding degradation on high-resource languages.[8]

Equivalent plots for the *zero-shot* setting are found in Figure 4.3. The effects of training steps and alpha are similar to the *few-shot* trends. However, the choice of vocabulary size does not have an obvious effect in this setting, an observation that is corroborated by our statistical analysis. Also of note is the fact that the performance for Skolt Sámi remains consistently poor across hyperparameters, which we investigate further in § 4.5.

4.4.3 Statistical analysis

Experimental Setup We conduct our regression analysis with linear mixed-effect models in the `lme4` package for R (Bates et al., 2015). LAPT steps and vocabulary size are treated as fixed continuous effects. Number of fine-tuning examples is also treated as a fixed continuous effect, but for the *few-shot* and *full-finetune* settings only. Task (POS vs UAS) is treated as a fixed categorical effect, following the observation that results for the two tasks mostly mirror each other, modulo a fixed offset (POS accuracy is higher than UAS). We justify this by testing a version of the regression with interaction terms between the task and other hyper-parameters (e.g. steps), but find no significant interactions. ANOVA confirms no significant difference from the model without task-interactions ($p = 0.95$).

Because the effect of α shows a different sign and magnitude between high- and low-resource languages, we model it as an *interaction* with a binary categorical variable representing whether the language is high- or low-resource. We justify the binary variable by the stark jump in resources between these two categories (see Section 4.3).

Finally, because of the complex factors leading to differing baseline performance between languages, we include a language-wise random-effect intercept. The final formula for this regression, as well as the full summary table with coefficients, can be found in the Appendix, Table C.6.

---

[8]Note: these plots for alpha are only representative of experiments with 100k steps, since longer-running experiments only tested α={0.1, 0.2}.

Few-shot / Full-finetune Results We find highly significant effects on performance ($p < 0.001$) for LAPT steps, vocabulary size, fine-tuning examples, and task.[9] Sampling alpha is significant in the low-resource case ($p = 0.035$), but not for high-resource languages ($p = 0.36$). This indicates choosing a lower alpha has a significant positive effect for low-resource language performance, without significantly hurting high-resource performance. The coefficient estimate for steps is 1.67, meaning an overall gain of 1.67 POS/UAS points for each 100k steps. The estimate for vocabulary size is 0.62 points per 16k tokens. The estimate for fine-tuning examples is 0.40 per 512 examples. In terms of our experiments, this means that doubling the number of steps from 100k to 200k is ~ 2.7 times as effective as doubling the vocabulary from 16k to 32k, and ~ 4.2 times as effective as doubling the number of fine-tuning examples to 1024. The estimate for alpha in the low-resource case is -1.36, meaning performance for low-resource languages drops about that much when alpha is raised from 0.1 to 0.2. Finally, we also test for, but find no significant interaction between, steps and vocabulary size; we confirm with ANOVA comparison that there is no significant difference between models with and without this interaction ($p = 0.43$).

Zero-shot Results Our regression for the *zero-shot* setting is similar to the previous, except that there is no variable for number of fine-tuning examples (which is not applicable for zero-shot transfer), and there is no interaction between sampling alpha and resource level, since all considered zero-shot languages are low-resource. The effects for steps and task are highly significant ($p < 0.001$); alpha is also significant ($p = 0.0027$). In contrast to the fine-tuned settings, vocabulary size is not significant ($p = 0.73$). The estimate for steps is 1.35 points per 100k steps. The estimate for alpha is -0.81 per increment of 0.1. These estimates are slightly smaller in magnitude than for the *few-shot/full-train* experiments; this could be partly due to the results for Skolt Sámi, which shows little change under any hyper-parameter configuration.

[9]Effect of task simply means baseline scores of each are different — about 14 points lower for UAS.

4.5 Discussion

Our discussion will first address the consistently poor performance seen on Skolt Sámi tasks (sms, §4.5.1). After this, we will move to the best practices suggested by our experimental results (§4.5.2).

4.5.1 Skolt Sámi error analysis

The consistently poor Skolt Sámi task performance across experimental settings suggests that the Sámi LAPT data may not be useful for this variant. We note that the datasets used for LAPT (in the case of Sámi, OPUS Tiedemann and Nygaard (2004) and the JHUBC McCarthy et al. (2020)) label most text as either undifferentiated Sámi (`se`) or as North Sámi (`sme`); however, Sámi is a group of languages, not all of which are mutually intelligible.

We therefore consider multiple tests for distribution shifts between the LAPT data and UD evaluation. The first is tokenizer efficiency, in characters per token. Our monolingual Sámi tokenizer trained on the LAPT data obtains 4.5 characters per token on that data, but this drops to 1.9 and 1.6 on the UD North Sámi and Skolt Sami datasets, respectively; this indicates a significant domain shift between the text seen in pre-training and in the UD datasets. We hypothesize that the model overcomes this vocabulary issue by available fine-tuning data for North Sámi, but that this does not occur for Skolt Sámi, since we evaluate it in a *zero-shot* setting.

In addition, the tokenizer shows a dramatic increase in OOV tokens when applied to Skolt Sámi — the unigram frequency for `<unk>` increases to 9%, from only 0.3% on the LAPT data.[10] Single-character tokens like `<õ>`, `<ä>`, `<â>`, and `<å>` also greatly increased in frequency, demonstrating the substantial hindrance that orthography differences can have on transfer between otherwise closely-related languages. These findings once again highlight importance of *quality* for language-modeling data, even when large web-scraped datasets have become the norm Kreutzer et al. (2022). Consequently, a future best practice may be to consider the intended downstream tasks (and their text distributions) when forming the vocabulary for a specialized multilingual model in order to minimize the occurrences of

[10]North Sámi OOV frequency is only 0.003%.

UNK tokens and facilitate better transfer learning between the language-modeling and task domains.

4.5.2 Best practices

Multilingualism is beneficial for many languages The baselines in §4.4.1 demonstrate that given an overall computational budget, it is more effective to adapt a multilingual model to jointly cover a group of languages than it is to adapt models for each individual language. This is especially true for low-resource languages, but surprisingly some high-resource languages like Hungarian and Russian also benefit from multilingual training. This supports the idea that multilingual training is useful for learning general patterns that are beneficial to the performance of many languages. Table 4.3 further shows that robust performance for low-resource languages like Komi and Moksha, which lack task fine-tuning sets, is only feasible with the combination of multilinguality and transfer learning.

Specialized vocab is more effective and efficient Our multilingual baselines in §4.4.1 demonstrate that even models with our smallest specialized vocabulary are on par with or outperform those retaining the large "cross-lingual" vocabulary from XLM-R, regardless of language. Table 4.6 shows that the 16k vocabulary tokenizes Uralic data with similar efficiency as the XLM-R vocabulary (in terms of mean sequence length), while yielding a model that is 35% of XLM-R's size. This reduction is significant both for the size of the model in disk/memory and for computational cost during training.[11]

Training steps vs. vocabulary size Our multi-variable regression analysis reveals that though both training steps and vocabulary size positively contribute to downstream performance in task fine-tuned settings, an additional 100k steps is almost three times as effective as adding 16k additional tokens (§4.4.3). It should be noted that increasing the vocabulary size from 16k to 32k only increases the number of floating point operations during training about 13% per token (for XLM-R base), while doubling the training steps doubles

[11] Per Kaplan et al. (2020), we estimate the number of operations per training step, per token as $6(N+dv+2d)$, where N is the number of non-embedding parameters, d is the hidden dimension, and v is the vocabulary size. Note this estimate is approximately proportional to the total number of parameters.

Vocab size	Parameters	Avg. length
16k	98.6M	49.9
32k	111.2M (+13%)	44.3 (-11%)
64k	136.4M (+23%)	39.7 (-10%)
128k	186.8M (+37%)	36.1 (-9%)
250k (orig)	278.3M	48.4

Table 4.6: Total number of model parameters and average sequence length for each vocabulary size. In parentheses are percent changes from the next-smallest vocabulary. Sequence length is computed on 100k sentences sampled from the LAPT set at $\alpha = 0.1$.

the number of operations. At the same time, a larger vocabulary reduces the tokenized sequence length, as the SentencePiece model becomes more efficient; shorter sequences lead to reduced computation.

However, as Table 4.6 shows, every doubling of the vocabulary size only reduces the average sequence length about 10%, so the parameter increase eventually outpaces efficiency from shorter sequences. Extra parameters also increase the model's memory footprint, which might in turn require more gradient accumulation steps to maintain a constant effective batch size on the same hardware; or it might make the model dependent on higher-tier hardware with more memory.

Finally, our regression analysis shows that vocabulary size does not have a significant effect on task performance in the *zero-shot* setting, which covers our lowest-resource languages (see §4.4.3 and Table 4.5). A best practice for adaptation to a low-resource language family might thus be to start with a relatively small vocabulary, and increase the size only until the increase in parameters outpaces the decrease in sequence length. Computational budget can then be spent on longer training rather than a larger model.

Lower alpha is better overall A key finding from our analysis is that sampling alpha values during multilingual training do not have a significant effect on task performance in high-resource languages, while low alphas *do* significantly benefit low-resource languages (§4.4.3). Our multilingual models thus frequently achieve their best average performance at the lower $\alpha = 0.1$, buoyed by the strong performance of low-resource languages.

This finding indicates that practitioners can aggressively up-sample lower-resource languages in multilingual datasets with little risk of degrading the performance of high-resource "anchor" languages. Further, as low as $\alpha = 0.1$, we see no evidence of "over-sampling" these low-resource languages harming downstream performance. However, we note that the high-resource languages we consider are in XLM-R's original pre-training set, which likely affects the model's robustness on those languages. Thus, it is an open question whether the dynamics of multilingual sampling are different in "from-scratch" training scenarios or in other high-resource, but previously unseen, languages.

4.6 Limitations

One limitation of our work is the small selection of evaluation tasks available for under-resourced languages. For most, the only high-quality datasets are found in expertly curated cross-lingual projects such as Universal Dependencies. While a few other datasets exist for under-resourced languages, they are often of questionable quality due to being automatically curated (Lignos et al., 2022). As such, our experiments are limited to POS tagging and UAS for dependency parsing.

Second, to maintain a feasible scope of work, we use only XLM-R as a base model for adaptation. Useful future work could include evaluating our adaptation techniques both in larger models, and for "generative" models trained with a traditional language modeling task rather than the masked language modeling employed by XLM-R. XGLM (Lin et al., 2022), for example, would be a natural next step, since it is both larger and generative. Evaluating multilingual generative models would also open the door to evaluations on more contemporary prompting-based tasks.

4.7 Conclusion

In this work, we show that adapting a pre-trained cross-lingual model to a language family is an effective method for greatly improving NLP task performance for languages in that family, especially those that are under-resourced. Multilingual adaptation soundly outperforms adaptation to single languages for all low-resource Uralic languages we test, as well as for half of the high-resource ones. Further, we show that specializing the model vocabulary for the Uralic family yields significant improvements over models that retain the large "cross-lingual" vocabulary of XLM-R, while simultaneously making the model much more computationally efficient and compact in disk/memory. Our statistical analysis of adaptation parameters reveals that both the number of LAPT steps and specialized vocabulary size have a significant positive effect on downstream task-finetuned performance. However, the language sampling alpha value is only significant for our low-resource languages, indicating that low alpha values can be chosen without significantly affecting high-resource language performance.

We therefore concur with Ogueji et al. (2021); Ogunremi et al. (2023); Chang et al. (2023); i.a. that *targeted* or *linguistically-informed* multilingual modeling is one of the most promising avenues for extending NLP advance to the majority of the world's languages. This approach both leverages the benefit of multilingualism for under-resourced languages and avoids the "Curse of Multilinguality" seen in massively-multilingual approaches. However, in view of the success of large pre-trained language models, and of the pre-training paradigm more generally (Gururangan et al., 2020), we propose that it is more effective to leverage transferable information in existing cross-lingual models, rather than training targeted models from scratch, as in these previous works. We hope that our findings will inform best practices for such targeted multilingual adaption when extending the benefits of pre-trained models to under-resource languages.

Chapter 5

CONCLUSION

We have presented three case studies for leveraging pre-trained models and targeted multilinguality for the benefit of NLP applications in under-resourced languages. In what remains, we summarize important points from these studies, and highlight broad, overarching principles for successful adaptation to such languages. We then close by briefly discussing our vision for integrating these principles into future work for under-resourced and endangered languages.

5.1 Summary and Discussion

In Chapter 2, we show that performance on a completely unsupervised morphological segmentation task can be transferred to a new language in few- and even zero-shot settings. This success is enabled by pooling raw data from a collection of typologically similar languages during pre-training, which are all under-resourced themselves. More broadly, we demonstrate that multilingual pre-training and transfer is possible with small models, very data-constrained settings, and a targeted selection of pre-training languages (rather than training on a broad, "massively multilingual" set).

While not all tasks require large model and data scale, we also demonstrate success in adapting large, pre-trained, cross-lingual models in order to leverage their useful language-general representations for under-resourced languages. In Chapter 3, we compare a wide selection of techniques for adapting a model's vocabulary to new languages. We demonstrate that completely throwing out pre-trained vocabulary representations and starting over from random initializations severely hinders the model's ability to adapt to a new vocabulary. However, even simple heuristics like matching new representations to the distribution of scripts in the pre-trained embedding space are adequate to give a "warm start" to adaptation through unsupervised language modeling. Further, this adaptation greatly reduces the

computational waste associated with training, using, and storing such models, significantly improving their accessibility outside of high-performance computing labs.

Finally, in Chapter 4 we provide a systematic analysis of the optimal methodology for adapting a pre-trained cross-lingual model to a linguistically informed set of languages such as a language family. We demonstrate that family-wise adaptation is a promising middle-ground between the extreme data scarcity of most individual languages on the one hand, and the "curse of multilinguality" on the other. Our models — adapted via multilingual language-modeling and vocabulary specialization — soundly outperform both monolingual and massively multilingual baselines. As in Chapter 3, the adaptation techniques we propose are far more computationally tractable than relying on large unmodified foundation models. For instance, our adaptation technique with reduced vocabulary size takes about 3 days to conduct 100,000 training steps on an Nvidia RTX 6000 GPU with 24GB of memory, whereas adapting the the original foundation model (XLM-R) takes over twice as long on an Nvidia Quadro RTX 8000, with twice as much memory (48GB).

5.2 Principles for Low-resource NLP

These findings suggest three broad principles for extending NLP applications to new, under-resourced languages. First: existing, robustly trained models and representations should be leveraged as a starting point where possible. Robustly training large models "from scratch" is intractable for most practitioners (Hoffmann et al., 2022; Kaplan et al., 2020), but the general-purpose representations from unsupervised pre-trained models are an invaluable starting point for a huge range of downstream languages and tasks (Conneau et al., 2020b; Peters et al., 2018, i.a.). Chapters 3 and 4 in particular demonstrate the advantage of using relevant pre-trained models and representations during adaptation to a new language, over alternatives trained from a random initialization. Chapter 2 also shows that where there is no existing "foundation model" appropriate for a certain task, even a relatively small amount of unsupervised pre-training can provide a reasonable substitute.

Second: components of pre-trained models that are not useful or optimized for new language domains can and should be substituted, or otherwise adapted. This is most plain from the results of our vocabulary specialization experiments in Chapters 3 and 4.

Models with vocabulary substitution perform as well or better than those that retain the original cross-lingual vocabulary, and do so with only 40% of the parameters of the original. This huge reduction in size (and computational complexity) is accomplished by simply dropping learned representations that are not useful for the language(s) of interest, while adapting the general-purpose model body for re-use via unsupervised training. Given the prohibitive costs of model training and inference on specialized machine learning hardware, these size reductions are a significant step towards accessibility, replicability, and robust experimentation for pre-trained multilingual models.

Third: targeted multilinguality constitutes a middle-ground between the intractability of training models for each individual language — most of which are severely under-resourced — and the diminishing returns of a massively multilingual approach, where interference between languages and the limits of model capacity lead to generally poor performance for all but a few languages. Multilingual modeling is a useful way to pool data and share language-general model parameters, but selectivity in which languages to group together allows for significantly reduced model sizes, and optimized performance for swaths of low-resource languages that are otherwise very poorly covered by existing models. This principle allows us to bootstrap morphological segmentation in extremely low-resource settings in Chapter 2 and to attain significant improvements over existing multilingual baselines for under-resourced Uralic languages in Chapter 4.

5.3 Future Work and Conclusion

The principles outlined here lead to several natural avenues for future work. For example, there may be the opportunity to extend advances in Large Language Model (LLM) engineering to many more languages than are currently covered. Current LLM research still tends to focus either on monolingual models (OpenAI et al., 2024) or on massively multilingual ones (Üstün et al., 2024). An alternative option for under-resourced languages could be to adapt an open-source LLM for either a linguistically or pragmatically related set of languages, such as a language family or the official/common languages of a nation.

Second, the documentation and revitalization of extremely under-resourced and endangered languages may benefit from moderately-sized, targeted multilingual models that can

be easily pre-trained and adapted via unsupervised methods (i.e. requiring only raw text). These models could form the core of a "toolkit" for quick adaptation to new languages, helping field linguists accomplish time-intensive tasks such as the transcription, phonetic alignment, and glossing of field recordings. For instance, a multilingually trained unsupervised segmentation model like the one presented in Chapter 2 could be deployed in tandem with an automatic phone recognition system to give a first-pass transcription of a language with which the linguist has limited familiarity.

For endangered and Indigenous languages especially, we believe this work to be of high urgency. Joshi et al. (2020b)'s survey of NLP resource availability classifies 2,460 languages — 98.65% of those surveyed, which are spoken by three billion people — as resource level 3 or below. In contrast, only 25 languages constitute levels 4 and 5. At the same time, UNESCO estimates that 90% of languages spoken at the beginning of the 21st century will be lost by the end of the century, listing "[lack of] response to new domains and media" as a key factor in language endangerment (Brenzinger et al., 2003). Given digital fluency is increasingly important to prosperity across the world, and that digital access is mediated by language technology (e.g. web search, assisted typing, translation), it is natural to postulate that this unequal access to language technology will likely drive language endangerment and loss. It is our hope that the work and principles in this thesis can be built upon by researchers and language advocates to close the enormous gap in language technology, bringing vital tools like web interfaces, assisted writing, reliable translation, speech recognition, and language-learning programs to the full diversity of the world's languages.

BIBLIOGRAPHY

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. Towards a cleaner document-oriented multilingual crawled corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.463.

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

Judit Ács, Dániel Lévai, and Andras Kornai. Evaluating transferability of BERT models on Uralic languages. In Flammie A Pirinen, Timofey Arhangelskiy, Trond Trosterud, and Michael Rießler, editors, *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*, pages 8–17, Syktyvkar, Russia (Online), September 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.iwclul-1.2.

Željko Agić and Ivan Vulić. JW300: A wide-coverage parallel corpus for low-resource languages. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1310. URL https://aclanthology.org/P19-1310.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 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, 2023.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL `https://aclanthology.org/2022.coling-1.382`.

Chantal Amrhein and Rico Sennrich. On Romanization for model transfer between scripts in neural machine translation. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2461–2469, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.223. URL `https://aclanthology.org/2020.findings-emnlp.223`.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL `https://aclanthology.org/2020.acl-main.421`.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. OpusFilter: A configurable parallel corpus filtering toolbox. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.20. URL `https://aclanthology.org/2020.acl-demos.20`.

Robert Austerlitz. Uralic languages. In Bernard Comrie, editor, *The World's Major Languages*, pages 477–483. Routledge, London, UK, 3 edition, 2008. ISBN 978-0-203-30152-4.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01.

Steven Bird. Sparse transcription. *Computational Linguistics*, 46(4):713–744, December 2020. doi: 10.1162/coli_a_00387. URL https://aclanthology.org/2020.cl-4.1.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL https://aclanthology.org/Q17-1010.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan

You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.

David Brambila. *Diccionario Rarámuri-castellano (Tarahumar).* Obra Nacional de la Buena Prensa, 1976.

Matthias Brenzinger, Arienne M. Dwyer, Tjeerd de Graaf, Colette Grinevald, Michael Krauss, Osahito Miyaoka, Nicholas Ostler, Osamu Sakiyama, María E. Villalón, Akira Y. Yamamoto, and Ofelia Zepeda. Language vitality and endangerment. UNESCO document, 2003.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Gina Bustamante, Arturo Oncevay, and Roberto Zariquiey. No data to crawl? monolingual corpus creation from PDF files of truly low-resource languages in Peru. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2914–2923, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://aclanthology.org/2020.lrec-1.356`.

Lyle Campbell, Terrence Kaufman, and Thomas C. Smith-Stark. Meso-America as a

Linguistic Area. *Language*, 62(3):530–570, 1986. ISSN 00978507, 15350665. URL http://www.jstor.org/stable/415477. Publisher: Linguistic Society of America.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. When Is Multilinguality a Curse? Language Modeling for 250 High- and Low-Resource Languages, November 2023. URL http://arxiv.org/abs/2311.09205. arXiv:2311.09205 [cs].

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. Parsing with multilingual BERT, a small corpus, and a small treebank. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.118. URL https://aclanthology.org/2020.findings-emnlp.118.

Luis Chiruzzo, Pedro Amarilla, Adolfo Ríos, and Gustavo Giménez Lugo. Development of a Guarani - Spanish parallel corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2629–2633, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.320.

J. Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical Multiscale Recurrent Neural Networks. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, 2017.

Alexis Conneau and Guillaume Lample. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada, 2019. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations.

In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL https://aclanthology.org/D18-1269.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. Emerging cross-lingual structure in pretrained language models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.536. URL https://aclanthology.org/2020.acl-main.536.

Adolfo Constenla, Feliciano Elizondo, and Francisco Pereira. *Curso Básico de Bribri*. Editorial de la Universidad de Costa Rica, 2004.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics, July 2002. doi: 10.3115/1118647.1118650. URL https://aclanthology.org/W02-0603.

Rubén Cushimariano Romano and Richer C. Sebastián Q. Ñaantsipeta asháninkaki birakochaki. Diccionario Asháninka-Castellano. Versión preliminar, 2008. URL http://www.lengamer.org/publicaciones/diccionarios/.

Carl de Marcken. Linguistic structure as composition and perturbation. In *34th Annual*

Meeting of the Association for Computational Linguistics, pages 335–341, Santa Cruz, California, USA, June 1996. Association for Computational Linguistics. doi: 10.3115/981863.981907. URL https://aclanthology.org/P96-1044.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, June 2021. doi: 10.1162/coli_a_00402. URL https://aclanthology.org/2021.cl-2.11.

Wietse de Vries and Malvina Nissim. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.74. URL https://aclanthology.org/2021.findings-acl.74.

Jacob Devlin. Multilingual BERT Readme, 2019. URL https://github.com/google-research/bert/blob/master/multilingual.md.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Konstantin Dobler and Gerard de Melo. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.829. URL https://aclanthology.org/2023.emnlp-main.829.

C. Downey, Shannon Drizin, Levon Haroutunian, and Shivin Thukral. Multilingual unsupervised sequence segmentation transfers to extremely low-resource languages. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5331–5346, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.366. URL https://aclanthology.org/2022.acl-long.366.

C.m. Downey, Fei Xia, Gina-Anne Levow, and Shane Steinert-Threlkeld. A masked segmental language model for unsupervised natural language segmentation. In Garrett Nicolai and Eleanor Chodroff, editors, *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 39–50, Seattle, Washington, July 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.sigmorphon-1.5. URL https://aclanthology.org/2022.sigmorphon-1.5.

C.m. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. In Duygu Ataman, editor, *Proceedings of the 3rd Workshop on Multilingual Representation Learning (MRL)*, pages 268–281, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.mrl-1.20. URL https://aclanthology.org/2023.mrl-1.20.

Timothy Dozat and Christopher D. Manning. Deep Biaffine Attention for Neural Dependency Parsing. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk95PK9le.

Abteen Ebrahimi and Katharina Kann. How to adapt your pretrained multilingual model to 1600 languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.351. URL https://aclanthology.org/2021.acl-long.351.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.435. URL `https://aclanthology.org/2022.acl-long.435`.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, April 1990. ISSN 0364-0213. doi: 10.1016/0364-0213(90)90002-E. URL `https://www.sciencedirect.com/science/article/pii/036402139090002E`.

Thomas Emerson. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005. URL `https://aclanthology.org/I05-3017`.

Fahim Faisal and Antonios Anastasopoulos. Phylogeny-inspired adaptation of multilingual models to new languages. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *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 (Volume 1: Long Papers)*, pages 434–452, Online only, November 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.aacl-main.34`.

I. Feldman and R. Coto-Solano. Neural Machine Translation Models with Back-Translation for the Extremely Low-Resource Indigenous Language Bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976, December 2020.

Sofía Flores Solórzano. Corpus Oral Pandialectal de la Lengua Bribri, 2017. URL `http://bribri.net`.

Ana-Paula Galarreta, Andrés Melgar, and Arturo Oncevay. Corpus Creation and Initial SMT Experiments between Spanish and Shipibo-konibo. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 238–244, Varna, Bulgaria, September 2017. INCOMA Ltd. doi: 10.26615/978-954-452-049-6_033. URL https://doi.org/10.26615/978-954-452-049-6_033.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. Towards continual learning for multilingual machine translation via vocabulary substitution. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.93. URL https://aclanthology.org/2021.naacl-main.93.

John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001. doi: 10.1162/089120101750300490. URL https://aclanthology.org/J01-2001.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, July 2009. ISSN 0010-0277. doi: 10.1016/j.cognition.2009.03.008. URL https://www.sciencedirect.com/science/article/pii/S0010027709000675.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In *Neural Networks*, volume 18, pages 602–610. Pergamon, July 2005.

Alex Graves, Fernández Santiago, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

Ximena Gutierrez-Vasques, Gerardo Sierra, and Isaac Hernandez Pompa. Axolotl: a web accessible parallel corpus for Spanish-Nahuatl. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4210–4214, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1666.

Thanh Le Ha, Jan Niehues, and Alexander Waibel. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, November 2016.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf.

Diego Huarcaya Taquiri. *Traducción Automática Neuronal para Lengua Nativa Peruana*. Bachelor's Thesis, Universidad Peruana Unión, 2020.

Carla Victoria Jara Murillo. *Gramática de la Lengua Bribri*. EDigital, 2018a. URL https://www.lenguabribri.com/gram%C3%A1tica-de-la-lengua-bribri.

Carla Victoria Jara Murillo. *I Ttè Historias Bribris*. Editorial de la Universidad de Costa Rica, 2 edition, 2018b. URL https://www.lenguabribri.com/i-tt%C3%A8-historias-bribris.

Carla Victoria Jara Murillo and Alí García Segura. *Se' ttö' bribri ie Hablemos en bribri*. EDigital, 2013. URL https://www.lenguabribri.com/se-tt%C3%B6-bribri-ie-hablemos-en-bribri.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5: 339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://aclanthology.org/Q17-1024.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020a. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 3 edition, 2024. URL https://web.stanford.edu/~jurafsky/slp3/ed3bookfeb3_2024.pdf.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza-Ruiz, and Hinrich Schütze. Fortification of neural morphological segmentation models for polysynthetic minimal-

resource languages. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1005. URL https://aclanthology.org/N18-1005.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, 2020. _eprint: 2001.08361.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to create and reuse words in open-vocabulary neural language modeling. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1492–1502, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1137. URL https://aclanthology.org/P17-1137.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to discover, ground and use words with segmental neural language models. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1645. URL https://aclanthology.org/P19-1645.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA, 2015.

Lingpeng Kong, Chris Dyer, and Noah A Smith. Segmental Recurrent Neural Networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*, San Juan, Puerto Rico, 2016. URL http://arxiv.org/abs/1511.06018.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72, 2022. doi: 10.1162/tacl_a_00447. URL `https://aclanthology.org/2022.tacl-1.4`.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL `https://aclanthology.org/D18-2012`.

William Lane, Mat Bettinson, and Steven Bird. A computational model for interactive transcription. In Eduard Dragut, Yunyao Li, Lucian Popa, and Slobodan Vucetic, editors, *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pages 105–111, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.dash-1.16. URL `https://aclanthology.org/2021.dash-1.16`.

Yann LeCun. Predictive learning. Keynote Talk, December 2016. 30th Conference on Neural Information Processing Systems.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed,

Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.813. URL https://aclanthology.org/2023.emnlp-main.813.

Constantine Lignos, Nolan Holley, Chester Palen-Michel, and Jonne Sälevä. Toward more meaningful resources for lower-resourced languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 523–532, 2022.

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. Few-shot learning with multilingual generative language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.616. URL https://aclanthology.org/2022.emnlp-main.616.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. Choosing transfer languages for cross-lingual learning. In

Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1301. URL https://aclanthology.org/P19-1301.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://aclanthology.org/2020.tacl-1.47.

Nikola Ljubešić, Vít Suchomel, Peter Rupnik, Taja Kuzman, and Rik van Noord. Language Models on a Diet: Cost-Efficient Development of Encoders for Closely-Related Languages via Additional Pretraining, April 2024. URL http://arxiv.org/abs/2404.05428. arXiv:2404.05428 [cs].

James Loriot, Erwin Lauriault, and Dwight Day. *Diccionario Shipibo-Castellano*. Ministerio de Educación, 1993.

Paul A. Luce. A computational analysis of uniqueness points in auditory word recognition. *Perception & Psychophysics*, 39(3):155–158, May 1986. ISSN 1532-5962. doi: 10.3758/BF03212485. URL https://doi.org/10.3758/BF03212485.

Manuel Mager, Diónico Carrillo, and Ivan Meza. Probabilistic Finite-State Morphological Segmenter for Wixarika (Huichol) Language. *Journal of Intelligent & Fuzzy Systems*, 34 (5):3081–3087, 2018.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In Manuel Mager, Arturo Oncevay, Annette Rios, Ivan Vladimir Meza Ruiz, Alexis Palmer, Graham Neubig, and Katharina Kann, editors, *Proceedings of the First*

Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 202–217, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.americasnlp-1.23. URL https://aclanthology.org/2021.americasnlp-1.23.

Enrique Margery. *Diccionario Fraseológico Bribri-Español Español-Bribri*. Editorial de la Universidad de Costa Rica, 2 edition, 2005.

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.352.

Sabrina Mielke and Jason Eisner. Spell Once, Summon Anywhere: A Two-Level Open-Vocabulary Language Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6843–6850, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33016843. URL https://ojs.aaai.org/index.php/AAAI/article/view/4660. Number: 01.

Elena Mihas. *Añaani katonkosatzi parenini, El idioma del alto Perené*. Milwaukee, WI: Clarks Graphics, 2011.

Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, Scottsdale, AR, USA, 2013.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, pages 3992–4006, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.293. URL https://aclanthology.org/2022.naacl-main.293.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li, editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1012.

Héctor Erasmo Gómez Montoya, Kervy Dante Rivas Rojas, and Arturo Oncevay. A continuous improvement framework of machine translation for Shipibo-konibo. In Alina Karakanta, Atul Kr. Ojha, Chao-Hong Liu, Jonathan Washington, Nathaniel Oco, Surafel Melaku Lakew, Valentin Malykh, and Xiaobing Zhao, editors, *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 17–23, Dublin, Ireland, August 2019. European Association for Machine Translation. URL https://aclanthology.org/W19-6804.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.38. URL https://aclanthology.org/2021.naacl-main.38.

Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*,

pages 875–880, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1103. URL https://aclanthology.org/D18-1103.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In Duygu Ataman, Alexandra Birch, Alexis Conneau, Orhan Firat, Sebastian Ruder, and Gozde Gul Sahin, editors, *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.11. URL https://aclanthology.org/2021.mrl-1.11.

Tolulope Ogunremi, Dan Jurafsky, and Christopher Manning. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1251–1266, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.93. URL https://aclanthology.org/2023.findings-eacl.93.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh,

Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan

Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

John E Ortega, Richard Alexander Castro-Mamani, and Jaime Rafael Montoya Samame. Overcoming resistance: The normalization of an Amazonian tribal language. In Alina Karakanta, Atul Kr. Ojha, Chao-Hong Liu, Jade Abbott, John Ortega, Jonathan Washington, Nathaniel Oco, Surafel Melaku Lakew, Tommi A Pirinen, Valentin Malykh, Varvara Logacheva, and Xiaobing Zhao, editors, *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 1–13, Suzhou, China, December 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.loresmt-1.1`.

Malte Ostendorff and Georg Rehm. Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning, January 2023. URL `http://arxiv.org/abs/2301.09626`. arXiv:2301.09626 [cs].

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1178. URL `https://aclanthology.org/P17-1178`.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276, 2021. doi: 10.1162/tacl_a_00365. URL `https://aclanthology.org/2021.tacl-1.16`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton

Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617. URL https://aclanthology.org/2020.emnlp-main.617.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. UNKs everywhere: Adapting multilingual language models to new scripts. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.800. URL https://aclanthology.org/2021.emnlp-main.800.

Prokopis Prokopidis, Vassilis Papavassiliou, and Stelios Piperidis. Parallel Global Voices: a collection of multilingual corpora with citizen media stories. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 900–905, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1144.

I. Richardson and F.M. Tyers. A morphological analyser for K'iche'. *Procesamiento de Lenguaje Natural*, 66:99–109, 2021.

Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, Singapore, 1989.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. URL https://aclanthology.org/2021.acl-long.243.

Kevin Scannell. The Crúbadán Project: Corpus building for under-resourced languages. *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop*, January 2007.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, and Et Alia. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model, 2023.

Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. ISSN 1053587X. doi: 10.1109/78.650093.

Lütfi Kerem Senel, Benedikt Ebing, Konul Baghirova, Hinrich Schuetze, and Goran Glavaš. Kardeş-NLU: Transfer to low-resource languages with the help of a high-resource cousin – a benchmark and evaluation for Turkic languages. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1672–1688, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.100.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Joongbo Shin, Yoonhyung Lee, Seunghyun Yoon, and Kyomin Jung. Fast and accurate deep bidirectional language representations for unsupervised learning. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 823–835, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.76. URL https://aclanthology.org/2020.acl-main.76.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. Transfer to a low-resource language via close relatives: The case study on Faroese. In Tanel Alumäe and Mark Fishel, editors, *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737, Tórshavn, Faroe Islands, May 2023. University of Tartu Library. URL https://aclanthology.org/2023.nodalida-1.74.

K. Song, X. Tan, Tao Qin, Jianfeng Lu, and T. Liu. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, CA, 2019.

Zhiqing Sun and Zhi-Hong Deng. Unsupervised neural word segmentation for Chinese via segmental language modeling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1531. URL https://aclanthology.org/D18-1531.

George Suárez. *The Mesoamerican Indian Languages*. Cambridge Language Surveys. Cambridge University Press, Cambridge, 1983.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, December 2006.

Jörg Tiedemann and Lars Nygaard. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva, editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2004/pdf/320.pdf`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

Estanislao A Txchajchal Batz, Luis Mateo Cúmez, and Candelaria Dominga López Ixcoy. *Gramática del Idioma K'iche'*. PLFM, 1996.

Francis Tyers and Robert Henderson. A corpus of k'iche' annotated for morphosyntactic structure. In Manuel Mager, Arturo Oncevay, Annette Rios, Ivan Vladimir Meza Ruiz, Alexis Palmer, Graham Neubig, and Katharina Kann, editors, *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 10–20, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.americasnlp-1.2. URL `https://aclanthology.org/2021.americasnlp-1.2`.

Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. Inducing word and part-of-speech with Pitman-Yor hidden semi-Markov models. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1774–1782, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1171. URL `https://aclanthology.org/P15-1171`.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. UDapter: Language adaptation for truly Universal Dependency parsing. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.180. URL https://aclanthology.org/2020.emnlp-main.180.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Ilia Polosukhin. Attention is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, Long Beach, CA, 2017. Neural Information Processing Systems Foundation. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. Sequence Modeling via Segmentations. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3674–3683, International Convention Centre, Sydney, Australia, August 2017. PMLR. URL http://proceedings.mlr.press/v70/wang17j.html.

Lihao Wang, Zongyi Li, and Xiaoqing Zheng. Unsupervised Word Segmentation with Bi-directional Neural Language Model. *arXiv:2103.01421 [cs]*, March 2021. URL http://arxiv.org/abs/2103.01421. arXiv: 2103.01421.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. Extending multilingual BERT to low-resource languages. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.240. URL https://aclanthology.org/2020.findings-emnlp.240.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing (EMNLP), pages 4438–4450, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.359. URL https://aclanthology.org/2020.emnlp-main.359.

Shijie Wu and Mark Dredze. Are all languages created equal in multilingual BERT? In Spandana Gella, Johannes Welbl, Marek Rei, Fabio Petroni, Patrick Lewis, Emma Strubell, Minjoon Seo, and Hannaneh Hajishirzi, editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.16. URL https://aclanthology.org/2020.repl4nlp-1.16.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Qingcheng Zeng, Lucas Garay, Peilin Zhou, Dading Chong, Yining Hua, Jiageng Wu, Yikang Pan, Han Zhou, Rob Voigt, and Jie Yang. GreenPLM: Cross-Lingual Transfer of Monolingual Pre-Trained Language Models at Almost No Cost. volume 6, pages 6290–6298, August 2023. doi: 10.24963/ijcai.2023/698. URL https://www.ijcai.org/proceedings/2023/698.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. Aya Model: An Instruction Finetuned Open-Access Multilingual Language Model, February 2024. URL http://arxiv.org/abs/2402.07827. arXiv:2402.07827 [cs].

Appendix A

APPENDIX TO CHAPTER 2

A.1 AmericasNLP Datasets

Composition The detailed composition of our preparation of the AmericasNLP 2021 training and validation sets can be found in Tables A.1 and A.2 respectively. `train_1.mono.cni`, `train_2.mono.cni`, `train_1.mono.shp`, and `train_2.mono.shp` are the additional monolingual sources for Asháninka and Shipibo-Konibo obtained from Bustamante et al. (2020). `train_downsampled.quy` is the version of the Quechua training set downsampled to 2^{15} lines to be more balanced with the other languages. `train.anlp` is the concatenation of the training set of every language before Quechua downsampling, and `train_balanced.anlp` is the version after Quechua downsampling. `train_downsampled.anlp` is the version of our multilingual set downsampled to be the same size as `train.quy`. MULTI-PT$_{full}$ is pre-trained on `train_balanced.anlp`, MULTI-PT$_{down}$ is pre-trained on `train_downsampled.anlp`, and QUECHUA-PT is pre-trained on `train.quy`.

Citations A more detailed description of the sources and citations for the AmericasNLP set can be found in the original shared task paper (Mager et al., 2021). Here, we attempt to give a brief listing of the proper citations.

All of the validation data originates from AmericasNLI (Ebrahimi et al., 2022) which is a translation of the Spanish XNLI set (Conneau et al., 2018) into the 10 languages of the AmericasNLP 2021 open task.

The training data for each of the languages comes from a variety of different sources. The **Asháninka** training data is sourced from Ortega et al. (2020); Cushimariano Romano and Sebastián Q. (2008); Mihas (2011) and consists of stories, educational texts, and environmental laws. The **Aymara** training data consists mainly of news text from the GlobalVoices corpus (Prokopidis et al., 2016) as available through OPUS (Tiedemann and

Nygaard, 2004). The **Bribri** training data is from six sources (Feldman and Coto-Solano, 2020; Margery, 2005; Jara Murillo, 2018a; Constenla et al., 2004; Jara Murillo and Segura, 2013; Jara Murillo, 2018b; Flores Solórzano, 2017) ranging from dictionaries and textbooks to story books. The **Guaraní** training data consists of blogs and web news sources collected by Chiruzzo et al. (2020). The **Nahuatl** training data comes from the Axolotl parallel corpus (Gutierrez-Vasques et al., 2016). The **Quechua** training data was created from the JW300 Corpus (Agić and Vulić, 2019), including Jehovah's Witnesses text and dictionary entries collected by Huarcaya Taquiri (2020). The **Rarámuri** training data consists of phrases from the Rarámuri dictionary (Brambila, 1976). The **Shipibo-Konibo** training data consists of translations of a subset of the Tatoeba dataset (Montoya et al., 2019), translations from bilingual education books (Galarreta et al., 2017), and dictionary entries (Loriot et al., 1993). The **Wixarika** training data consists of translated Hans Christian Andersen fairy tales from Mager et al. (2018).

No formal citation was given for the source of the **Hñähñu** training data (see Mager et al., 2021).

A.2 Hyperparameter Details

Pre-training The character embeddings for our multilingual model are initialized by training CBOW (Mikolov et al., 2013) on the AmericasNLP training set for 32 epochs, with a window size of 5. Special tokens like `<bos>` that do not appear in the training corpus are randomly initialized. These pre-trained embeddings are not frozen during training.

We pre-train for 16,768 steps, using the Adam optimizer (Kingma and Ba, 2015). We apply a linear warmup for 1024 steps, and a linear decay afterward. We sweep eight learning rates on a grid of the interval $[0.0005, 0.0009]$ and encoder dropout values $\{12.5\%, 25\%\}$. A dropout rate of 6.25% is applied both to the embeddings before being passed to the encoder, and to the hidden-state and start-symbol encodings input to the decoder (see Downey et al., 2022b). Checkpoints are taken every 128 steps.

K'iche' Transfer Experiments Similar to the pre-trained model, character embeddings are initialized using CBOW on the given training set for 32 epochs with a window size of 5,

and these embeddings are not frozen during training.

All models are trained using the Adam optimizer (Kingma and Ba, 2015) for 8192 steps on all but the two smallest sizes, which are trained for 4096 steps. A linear warmup is used for the first 1024 steps (512 for the smallest sets), followed by linear decay. We set the maximum segment length to 10. A dropout rate of 6.25% is applied to the input embeddings, plus h and the start-symbol for the decoder. Checkpoints are taken every 64 steps for sizes 256 and 512, and every 128 steps for every other size.

For all training set sizes, we sweep 5 learning rates and 3 encoder dropout rates, but the swept set is different for each. For size 256, we sweep learning rates {5e-5, 7.5e-5, 1e-4, 2.5e-4, 5e-4} and (encoder) dropout rates {12.5%, 25%, 50%}. For size 2048, we sweep learning rates {1e-4, 2.5e-4, 5e-4, 7.5e-4, 1e-3} and dropouts {12.5%, 25%, 50%}. For the full training size, we sweep learning rates {1e-4, 2.5e-4, 5e-4, 7.5e-4, 1e-3} and dropouts {6.5%, 12.5%, 25%}.

Language	File	Lines	Total Tokens	Unique Tokens	Total Characters	Unique Characters	Mean Token Length
All	train.anlp	259,207	2,682,609	400,830	18,982,453	253	7.08
All	train_balanced.anlp	171,830	1,839,631	320,331	11,981,011	241	6.51
All	train_downsampled.anlp	120,145	1,284,440	255,392	8,365,710	221	6.51
Asháninka	train.cni	3,883	26,096	12,490	232,494	65	8.91
Asháninka	train_1.mono.cni	12,010	99,329	27,963	919,897	48	9.26
Asháninka	train_2.mono.cni	593	4,515	2,325	42,093	41	9.32
Aymara	train.aym	6,424	96,075	33,590	624,608	156	6.50
Bribri	train.bzd	7,508	41,141	7,858	167,531	65	4.07
Guaraní	train.gug	26,002	405,449	44,763	2,718,442	120	6.70
Hñähñu	train.oto	4,889	72,280	8,664	275,696	90	3.81
Nahuatl	train.nah	16,684	351,702	53,743	1,984,685	102	5.64
Quechua	train.quy	120,145	1,158,273	145,899	9,621,816	114	8.31
Quechua	train_downsampled.quy	32,768	315,295	64,148	2,620,374	95	8.31
Rarámuri	train.tar	14,720	103,745	15,691	398,898	74	3.84
Shipibo Konibo	train.shp	14,592	62,850	17,642	397,510	56	6.32
Shipibo Konibo	train_1.mono.shp	22,029	205,866	29,534	1,226,760	61	5.96
Shipibo Konibo	train_2.mono.shp	780	6,424	2,618	39,894	39	6.21
Wixarika	train.hch	8,948	48,864	17,357	332,129	67	6.80

Table A.1: Composition of the AmericasNLP 2021 training sets

Language	File	Lines	Total Tokens	Unique Tokens	Total Characters	Unique Characters	Mean Token Length
All	dev.anlp	9,122	79,901	27,597	485,179	105	6.07
Asháninka	dev.cni	883	6,070	3,100	53,401	63	8.80
Aymara	dev.aym	996	7,080	3,908	53,852	64	7.61
Bribri	dev.bzd	996	12,974	2,502	50,573	73	3.90
Guaraní	dev.gug	995	7,191	3,181	48,516	70	6.75
Hñähñu	dev.oto	599	5,069	1,595	22,712	69	4.48
Nahuatl	dev.nah	672	4,300	1,839	31,338	56	7.29
Quechua	dev.quy	996	7,406	3,826	58,005	62	7.83
Rarámuri	dev.tar	995	10,377	2,964	55,644	48	5.36
Shipibo Konibo	dev.shp	996	9,138	3,296	54,996	65	6.02
Wixarika	dev.hch	994	10,296	3,895	56,142	62	5.45

Table A.2: Composition of the AmericasNLP 2021 validation sets

Appendix B

APPENDIX TO CHAPTER 3

B.1 Data Details

General information about the language data used in this study can be found in Table B.1. All training data used in our experiments is cleaned and deduplicated using the OpusFilter package (Aulamo et al., 2020). For the lowest-resource languages (Erzya and Sami) we additionally filter out lines that are identified as English with a probability of 90% or higher, since positive automatic language-identification for low-resource languages is likely not robust (Kreutzer et al., 2022). We additionally filter out lines composed of less than 2 tokens, lines with an average token length of greater than 16 characters, lines with tokens longer than 32 characters, and lines composed of fewer than 50% alphabetic characters.

For POS tagging evaluation, most languages have a standard train/dev/test split curated the original Universal Dependencies dataset (de Marneffe et al., 2021). Erzya, however, only has a standard train/test split. To form a dev split, we randomly sample 300 sentences from the train split. The WikiAnn dataset (Pan et al., 2017) does not ship with standard train/dev/test splits, so we create random 85/5/10% splits of each language for this purpose, with a minimum dev/test size of 256 and 512 sentences respectively.

B.2 Training Details

The main details of our experimental process can be found in § 3.4. Here we provide our choice of hyperparameters and other details relevant to reproducibility. The code used to run all experiments can be found at `github.com/cmdowney88/EmbeddingStructure`. All models are trained and fine-tuned on Nvidia Quadro RTX 6000 GPUs using the Adam optimizer (Kingma and Ba, 2015).

Hyperparameters for Language-Adaptive Pre-Training (LAPT) can be found in Table B.2. If NaN losses were encountered during training, `max_gradient_norm` was reduced to 0.5.

Language	Code	Family	Script	XLM-R Data (GB)	LAPT Data (GB)
Armenian	hy	Indo-European	Armenian	5.5	1.2
Basque	eu	isolate	Latin	2.0	0.35
Erzya	myv	Uralic	Cyrillic	0	0.006
Estonian	et	Uralic	Latin	6.1	3.0
Finnish	fi	Uralic	Latin	54.3	9.1
Hebrew	he	Afro-Asiatic	Hebrew	31.6	7.7
Hungarian	hu	Uralic	Latin	58.4	13.0
Russian	ru	Indo-European	Cyrillic	278.0	10.0
Sami	sme	Uralic	Latin	0	0.004
Telugu	te	Dravidian	Telugu	4.7	0.9

Table B.1: Training data breakdown by language. XLM-R data is the amount of data used in the pre-training of that model. LAPT data is the amount used for training in our current experiments, after cleaning/deduplicating.

For multilingual sampling during training, each language's training data is capped at approximately 2GB.

Hyperparameters for task fine-tuning on POS and NER are in Table B.3. For NER, the reported evaluation metric is entity-wise F1, meaning tokens with label `O` are ignored. In order to prevent models from learning to output only the majority class `O` during training, the loss for the `O` tokens in each batch is down-weighted to have the same influence as the tokens that actually correspond to a named entity. We cap fine-tuning training data at 32,768 sequences.

B.3 Uralic Results

The results for multilingual adaptation to the Uralic family can be found in Tables B.4 and B.5. These results mostly follow the trends discussed in § 3.5 (LAPT-EMB consistently underperforms LAPT-FULL, off-the-shelf performance is best for high-resource languages,

Hyperparameter	Value
`mlm_masking_prob`	0.15
`max_sequence_length`	256
`learning_rate`	1e-5
`lr_schedule`	linear
`batch_size`	200
`max_gradient_norm`	1.0

Table B.2: Hyperparameters for model training (LAPT)

LAPT with full cross-lingual vocab performs marginally better than other methods). It should be noted that for both Erzya and Hungarian, the best POS accuracy is achieved with SCRIPT+POSN+IDENT initialization (better even than LAPT with the fully cross-lingual vocabulary). Results for the very low-resource language Erzya are generally higher than with multilingual training on unrelated languages, which could suggest a benefit to training with closely-related languages. This observation does not clearly hold for Sami (the other very low-resource language), however. Note that Russian is not a Uralic language — we include it for multilingual training in order to robustly train embeddings for the Cyrillic script, in which Erzya is written. Erzya is also spoken primarily within the Russian Federation, making loan-words likely.

Hyperparameter	Value
`max_sequence_length`	256
`learning_rate`	5e-6
`lr_schedule`	constant
`max_epochs`	64
`eval_interval` (epochs)	2
`patience` (epochs)	8 (POS) / 4 (NER)
`batch_size`	72
`max_gradient_norm`	1.0

Table B.3: Hyperparameters for model task fine-tuning

LAPT	REINIT	Erzya	Estonian	Finnish	Hungarian	North Sami	Russian	Avg
*	*	56.3 ± 5.3	95.6 ± 0.1	97.5 ± 0.1	93.7 ± 1.5	71.2 ± 1.8	98.6 ± 0.1	85.9
FULL	*	72.5 ± 2.6	95.8 ± 0.1	97.7 ± 0.2	94.1 ± 1.9	82.9 ± 0.4	98.6 ± 0.04	90.3
FULL	FOCUS+IDENT	**73.8 ± 2.7**	**95.3 ± 0.2**	**97.2 ± 0.1**	92.5 ± 1.6	**80.1 ± 1.4**	**98.4 ± 0.04**	**89.6**
FULL	SCRIPT+POSN+IDENT	**73.0 ± 1.4**	94.7 ± 0.3	96.6 ± 0.1	**94.8 ± 0.7**	**78.0 ± 2.3**	**98.4 ± 0.01**	89.3
FULL	SCRIPT+IDENT	67.7 ± 11.0	94.3 ± 0.3	96.4 ± 0.1	**94.7 ± 0.7**	**78.8 ± 2.2**	**98.4 ± 0.03**	88.4
FULL	SCRIPT+POSN	71.2 ± 2.7	88.7 ± 0.4	90.6 ± 0.1	86.8 ± 0.4	72.9 ± 2.0	97.2 ± 0.02	84.7
FULL	SCRIPT	65.9 ± 4.6	85.6 ± 1.3	89.1 ± 0.3	85.2 ± 0.2	73.5 ± 1.6	96.9 ± 0.05	82.7
FULL	IDENT	59.8 ± 1.2	92.2 ± 0.03	95.2 ± 0.04	91.8 ± 2.8	68.9 ± 0.9	98.2 ± 0.03	84.3
FULL	RANDOM	53.7 ± 3.2	71.9 ± 0.6	73.1 ± 0.2	59.6 ± 1.6	63.9 ± 0.9	84.9 ± 1.9	67.8
EMB	FOCUS+IDENT	**66.3 ± 1.2**	**94.7 ± 0.1**	**96.8 ± 0.2**	**94.2 ± 0.8**	**73.3 ± 1.6**	**98.4 ± 0.05**	**87.3**
EMB	SCRIPT+POSN+IDENT	64.2 ± 2.8	93.0 ± 0.1	95.5 ± 0.03	**93.6 ± 0.8**	**72.7 ± 2.6**	98.3 ± 0.05	86.2
EMB	SCRIPT+IDENT	55.8 ± 4.1	92.8 ± 0.2	95.4 ± 0.04	92.3 ± 1.6	69.8 ± 1.6	98.3 ± 0.04	84.1
EMB	SCRIPT+POSN	54.5 ± 4.3	74.2 ± 0.8	79.5 ± 0.7	62.1 ± 2.6	65.2 ± 2.0	94.8 ± 0.4	71.7
EMB	SCRIPT	48.7 ± 0.04	56.9 ± 15.6	71.6 ± 3.2	54.3 ± 4.4	58.0 ± 1.7	91.4 ± 1.8	63.5
EMB	IDENT	49.2 ± 1.7	90.6 ± 0.4	94.4 ± 0.03	84.8 ± 2.9	64.7 ± 1.3	97.9 ± 0.1	80.3
EMB	RANDOM	48.6 ± 0.2	64.5 ± 4.1	66.4 ± 1.2	43.6 ± 0.1	45.8 ± 4.2	84.0 ± 1.4	58.8

Table B.4: Uralic family multilingual LAPT: POS tagging accuracy after fine-tuning

LAPT	REINIT	Erzya	Estonian	Finnish	Hungarian	Russian	Avg
*	*	89.5 ± 0.6	93.3 ± 0.2	90.7 ± 0.1	92.4 ± 0.1	90.9 ± 0.2	91.4
FULL	*	90.5 ± 0.5	93.8 ± 0.2	91.0 ± 0.2	92.4 ± 0.3	91.0 ± 0.2	91.8
FULL	FOCUS+IDENT	**89.4 ± 1.7**	**92.5 ± 0.1**	**89.8 ± 0.2**	**91.2 ± 0.4**	**90.4 ± 0.1**	**90.7**
FULL	SCRIPT+POSN+IDENT	88.7 ± 0.5	92.2 ± 0.4	89.2 ± 0.2	90.9 ± 0.2	90.1 ± 0.1	90.2
FULL	SCRIPT+IDENT	**89.3 ± 0.4**	**92.7 ± 0.3**	89.2 ± 0.4	**91.3 ± 0.1**	90.0 ± 0.2	90.5
FULL	SCRIPT+POSN	**89.5 ± 1.0**	87.9 ± 0.2	84.2 ± 0.3	86.3 ± 0.3	86.2 ± 0.2	86.8
FULL	SCRIPT	**88.9 ± 0.8**	87.5 ± 0.3	83.3 ± 0.1	86.3 ± 0.2	85.5 ± 0.1	86.3
FULL	IDENT	81.1 ± 0.8	91.6 ± 0.1	88.2 ± 0.2	90.7 ± 0.3	89.6 ± 0.1	88.2
FULL	RANDOM	73.7 ± 2.7	53.1 ± 30.7	0.0 ± 0.0	32.9 ± 33.0	65.1 ± 2.2	45.0
EMB	FOCUS+IDENT	**88.6 ± 0.6**	**92.4 ± 0.3**	**89.6 ± 0.1**	**91.1 ± 0.1**	**90.0 ± 0.1**	**90.3**
EMB	SCRIPT+POSN+IDENT	86.6 ± 1.1	91.4 ± 0.2	88.8 ± 0.3	90.5 ± 0.2	89.9 ± 0.1	89.4
EMB	SCRIPT+IDENT	87.0 ± 1.3	91.8 ± 0.1	88.6 ± 0.3	**91.0 ± 0.2**	89.6 ± 0.2	89.6
EMB	SCRIPT+POSN	85.0 ± 1.2	84.2 ± 0.4	78.1 ± 0.3	81.9 ± 0.5	82.1 ± 0.2	82.3
EMB	SCRIPT	82.9 ± 2.6	82.4 ± 1.3	72.5 ± 1.3	80.7 ± 0.4	79.0 ± 0.2	79.5
EMB	IDENT	71.0 ± 4.4	90.1 ± 0.3	87.0 ± 0.4	89.9 ± 0.2	88.7 ± 0.1	85.3
EMB	RANDOM	64.9 ± 1.9	0.0 ± 0.0	13.6 ± 23.5	0.0 ± 0.0	54.4 ± 2.2	26.6

Table B.5: Uralic family multilingual LAPT: entity-wise NER F1 score after fine-tuning. A score of 0.0 results from the model learning to output only class `O` (not a named entity) which is the majority class. Sami does not have enough NER data for fine-tuning

Appendix C

APPENDIX TO CHAPTER 4

C.1 Training Details

The main details of our experimental process can be found in § 4.3. Here we provide our choice of hyperparameters and other details relevant to reproducibility.

C.1.1 Data

All LAPT data used in our experiments is cleaned and de-duplicated with the OpusFilter package (Aulamo et al., 2020). For low-resource languages, we additionally filter out lines that are identified as English with a probability of 90% or higher, since positive automatic language-identification for low-resource languages is likely not robust (Kreutzer et al., 2022). We additionally filter out lines composed of less than 2 tokens, lines with an average token length of greater than 16 characters, lines with tokens longer than 32 characters, and lines composed of fewer than 50% alphabetic characters. We reserve 5% of the total LAPT data in each language for a development set, and 5% for a test set.

C.1.2 Parameters

All models are trained and fine-tuned on Nvidia Quadro RTX 6000 GPUs using the Adam optimizer (Kingma and Ba, 2015). Hyperparameters for Language-Adaptive Pre-Training (LAPT) can be found in Table C.1.

C.2 Evaluation Details

C.2.1 Data

Most language have a standard train/dev/test split curated the original Universal Dependencies dataset (de Marneffe et al., 2021). Erzya, however, only has a standard train/test split.

Hyperparameter	Value
`mlm_masking_prob`	0.15
`max_sequence_length`	256
`learning_rate`	1e-5
`lr_schedule`	linear
`batch_size`	200
`max_gradient_norm`	1.0

Table C.1: Hyperparameters for model training (LAPT)

To form a dev split, we randomly sample 300 sentences from the train split. The inventory of UD evaluation data can be found in Table C.2.

C.2.2 Parameters

Hyperparameters for task fine-tuning on POS and UAS are in Table C.3. We cap fine-tuning training data at 32,768 sequences (only relevant for Russian).

C.2.3 Unlabeled Attachment Score

Unlabeled Attachment Score (UAS) is the accuracy with which a model assigns each word its proper dependency head. Our implementation uses the graph biaffine algorithm defined in Dozat and Manning (2017). The contextual embedding representation for each token r_i is passed through each of two feed-forward layers, to produce a representation of this token as a head and as a dependent, respectively:

$$h_i^{head} = \text{FFN}^{head}(r_i)$$

$$h_i^{dep} = \text{FFN}^{dep}(r_i)$$

The score of a directed edge i $\rightarrow$ j, is then assigned according to a biaffine scoring function:

$$\text{Biaffine}(h_i^{head}, h_j^{dep}) = U_{\text{arc}} + W_{\text{arc}} + b$$

Language	Code	Branch	Script	Train	Dev	Test
Russian	ru	n/a	Cyrillic	69,630	8,906	8,800
Finnish	fi	Finnic	Latin	14,981	1,875	1,867
Estonian	et	Finnic	Latin	5,444	833	913
North Sámi	sme	Sámi	Latin	2,001	256	865
Hungarian	hu	Hungarian	Latin	910	441	449
Erzya	myv	Mordvinic	Cyrillic	896	300	921
Komi	koi	Permic	Cyrillic	0	0	663
Moksha	mdf	Mordvinic	Cyrillic	0	0	446
Skolt Sámi	sms	Sámi	Latin	0	0	244
Karelian	krl	Finnic	Latin	0	0	228
Livvi	olo	Finnic	Latin	0	0	106

Table C.2: Universal Dependencies evaluation set sizes, by number of examples (sentences)

$$U_{\text{arc}} = h_j^{dep} \cdot U_{\text{arc\_head}}^T$$

$$U_{\text{arc\_head}} = U \cdot h_i^{head}$$

$$W_{\text{arc}} = W \cdot h_i^{head}$$

where U, W, and b are weights learned by the model. A probability distribution over possible heads is then computed by passing score(i $\rightarrow$ j) through a softmax layer. Our implementation is based on Jurafsky and Martin (2024) and `https://www.cse.chalmers.se/~richajo/nlp2019/l7/Biaffine%20dependency%20parsing.html`.

C.3 Additional results

Results and visualizations for the POS task can be found in this appendix. For POS, the multilingual baseline without vocabulary specialization performs more on-par with models with specialized vocabulary (Tables C.4, C.5). This is possibly due to the relative simplicity

Hyperparameter	Value
`max_sequence_length`	256
`learning_rate`	5e-6
`lr_schedule`	constant
`max_epochs`	64
`eval_interval` (epochs)	2
`patience` (epochs)	none / 8
`batch_size`	72
`max_gradient_norm`	1.0

Table C.3: Hyperparameters for model task fine-tuning. *few-shot* has no early stopping. *Full-finetune* and *zero-shot* settings have early stopping after patience of 8 epochs

of the task. The parameter-wise trends for POS are mostly the same as for UAS (Figures C.1, C.2).

C.4 Regression tables

The full regression summaries from the `lme4` package (Bates et al., 2015) can be found in Tables C.6-C.9. These cover both the fine-tuned (*few-shot*/*full-finetune*) and *zero-shot* models. As mentioned in § 4.3, we test four values of alpha for experiments with 100k steps, but only two values for longer experiments. Because this introduces artificial correlation of input variables, we separate the regression with two alphas as our "main" results, but include the summary of regressions with four values (but no variation in training steps) here (Tables C.7 and C.9). These secondary regressions show a greater effect size for low-resource alpha, indicating the estimate between the alpha values 0.1 and 0.2 might not accurate estimate the larger trends. Note that these secondary regressions do not change the standings of which variables are significant.

LAPT	Alpha	Vocab	Erzya	North Sámi	Estonian	Finnish	Hungarian	Russian	Avg
0	*	250k (orig)	50.9 ± 1.9	53.8 ± 3.1	63.9 ± 5.4	66.7 ± 3.7	81.5 ± 5.4	86.8 ± 1.0	67.3
400k	0.1	250k (orig)	75.2 ± 2.6	**77.2 ± 2.6**	**84.2 ± 0.3**	**83.3 ± 2.1**	88.0 ± 3.2	**90.1 ± 2.0**	82.7
400k	0.1	16k	76.1 ± 3.3	73.2 ± 1.2	77.7 ± 3.9	79.7 ± 2.6	89.3 ± 1.3	87.5 ± 0.5	80.6
400k	0.1	32k	72.3 ± 4.2	71.4 ± 1.2	82.7 ± 2.4	82.3 ± 3.8	87.7 ± 2.4	88.0 ± 2.2	80.7
400k	0.1	64k	**78.0 ± 1.4**	**76.5 ± 3.5**	83.0 ± 2.4	**85.4 ± 2.2**	**94.1 ± 1.1**	88.1 ± 1.5	**84.2**

Table C.4: Few-shot POS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocabulary replacement). Second row is XLM-R with original cross-lingual vocabulary, but fine-tuned on Uralic languages with LAPT

LAPT	Alpha	Vocab	Karelian	Komi	Livvi	Moksha	Skolt Sámi	Avg
0	*	250k (orig)	77.7 ± 0.6	49.6 ± 0.6	73.7 ± 0.8	64.4 ± 0.3	55.0 ± 1.2	64.1
400k	0.1	250k (orig)	86.7 ± 0.2	80.0 ± 0.2	85.2 ± 0.4	79.4 ± 0.2	**56.1 ± 1.0**	**77.5**
400k	0.1	16k	**87.7 ± 0.2**	80.0 ± 0.3	85.0 ± 0.2	78.3 ± 0.2	**55.4 ± 0.3**	77.3
400k	0.1	32k	87.3 ± 0.3	80.1 ± 0.2	**85.6 ± 0.4**	78.6 ± 0.5	53.7 ± 0.3	77.0
400k	0.1	64k	87.4 ± 0.4	**81.4 ± 0.4**	**85.6 ± 0.2**	**79.6 ± 0.1**	52.2 ± 1.7	77.2

Table C.5: Zero-shot POS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocabulary replacement). Second row is XLM-R with original cross-lingual vocabulary, but fine-tuned on Uralic languages with LAPT

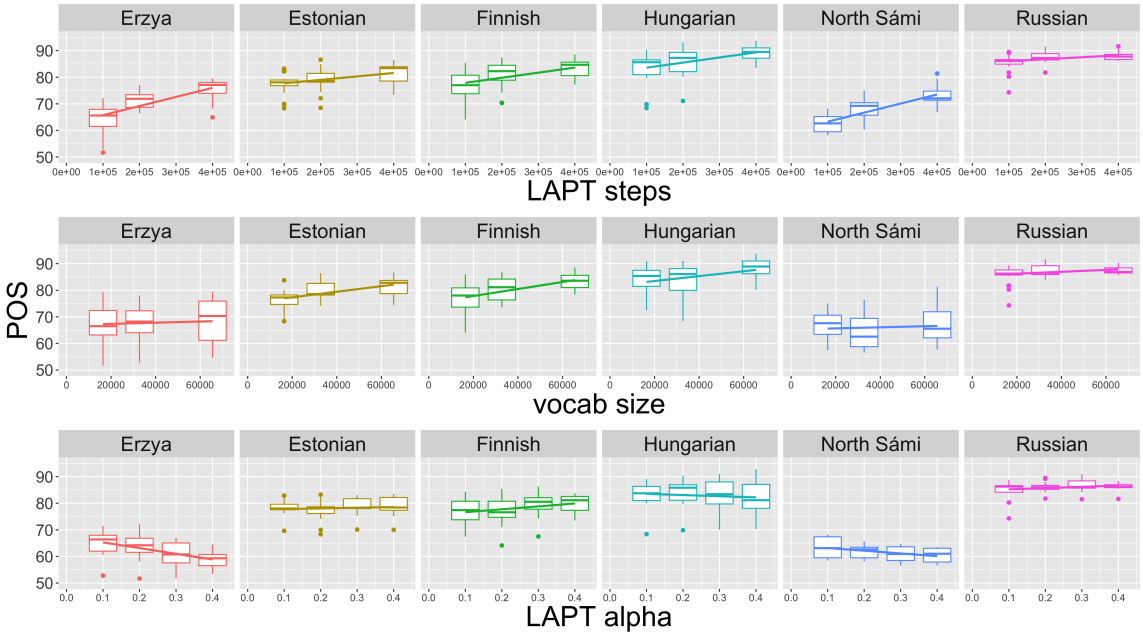


Figure C.1: Few-shot POS — effect of hyper-parameters by language, marginalized across other parameter settings

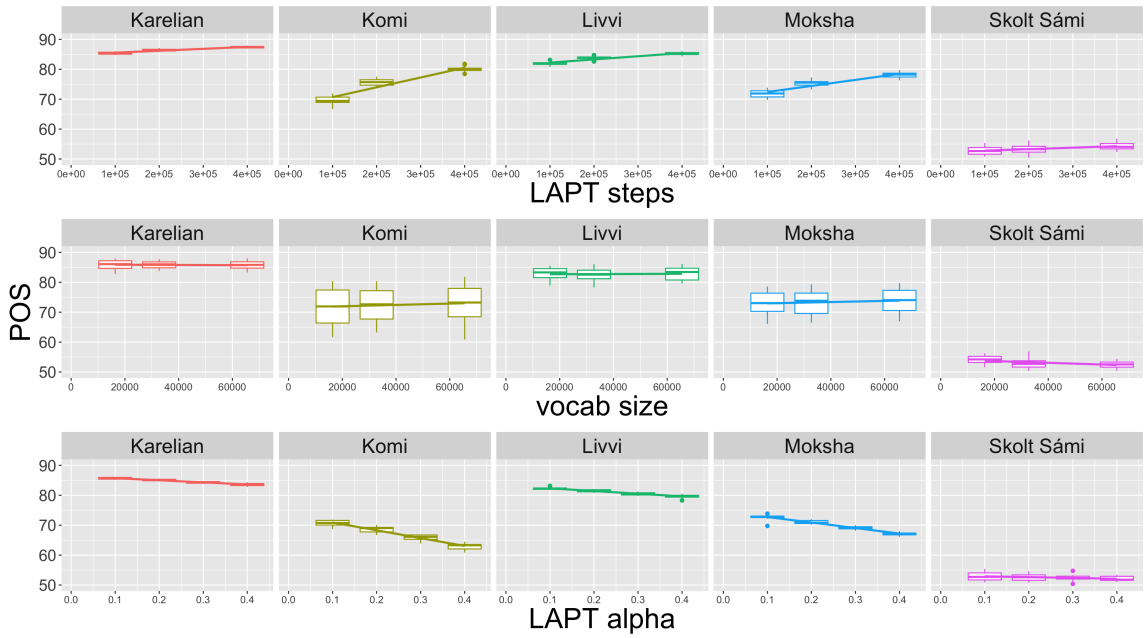


Figure C.2: Zero-shot POS — effect of hyper-parameters by language, marginalized across other parameter settings

Fixed effects	Estimate	Std. Errror	df	t value	p value
`(Intercept)`	**75.93**	2.53	5.63	29.97	**2.00e-07**
`lapt_steps`	**1.67**	0.15	1691.67	11.16	< **2e-16**
`vocab_size`	**0.62**	0.15	1691.67	4.15	**3.49e-05**
`finetuning_lines`	**0.40**	0.01	1696.77	30.32	< **2e-16**
`taskuas`	**-13.84**	0.38	1691.67	-36.71	< **2e-16**
`resourcehigh:lapt_alpha`	0.42	0.46	1582.98	0.92	0.3606
`resourcelow:lapt_alpha`	**-1.36**	0.64	1239.05	-2.11	**0.0347**

Table C.6: Regression summary table for *few-shot* and *full-finetune* settings. Significant coefficients and p values in bold. This regression covers all training lengths (step numbers), but only includes alphas {0.1, 0.2}. Formula:
`lmer(accuracy ~ lapt_steps + vocab_size + finetuning_lines + task + resource:lapt_alpha + (1 | language))`

Fixed effects	Estimate	Std. Errror	df	t value	p value
`(Intercept)`	**78.39**	2.95	5.39	26.61	**6.27e-07**
`vocab_size`	**0.39**	0.19	1140.76	2.01	**0.0448**
`finetuning_lines`	**0.42**	0.02	1146.00	25.14	< **2e-16**
`taskuas`	**-14.16**	0.48	1140.76	-29.44	< **2e-16**
`resourcehigh:lapt_alpha`	0.19	0.26	1132.87	0.72	0.4730
`resourcelow:lapt_alpha`	**-2.38**	0.37	1058.70	-6.45	**1.66e-10**

Table C.7: Secondary regression summary table for *few-shot* and *full-finetune* settings. Significant coefficients and p values in bold. This regression covers all values of alpha {0.1, 0.2, 0.3, 0.4}, which are only tested in experiments with 100k training steps. Thus, the `lapt_steps` variable is excluded from this regression. Formula:
`lmer(accuracy ~ vocab_size + finetuning_lines + task + resource:lapt_alpha + (1 | language))`

Fixed effects	Estimate	Std. Errror	df	t value	p value
`(Intercept)`	**72.68**	5.20	4.09	13.99	**1.31e-4**
`lapt_steps`	**1.35**	0.11	711.00	12.58	< **2e-16**
`vocab_size`	0.04	0.11	711.00	0.35	0.7266
`lapt_alpha`	**-0.81**	0.27	711.00	-3.02	**2.66e-3**
`taskuas`	**-12.89**	0.27	711.00	-48.02	< **2e-16**

Table C.8: Regression summary table for *zero-shot* setting. Significant coefficients and p values in bold. This regression covers all training lengths (step numbers), but only includes alphas {0.1, 0.2}. Formula:
`lmer(accuracy ~ lapt_steps + vocab_size + lapt_alpha + task + (1 | language))`

Fixed effects	Estimate	Std. Errror	df	t value	p value
`(Intercept)`	**74.33**	4.72	4.08	15.73	**8.31e-5**
`vocab_size`	-0.05	0.12	472.00	-0.38	0.7020
`lapt_alpha`	**-1.30**	0.14	472.00	-9.46	< **2e-16**
`taskuas`	**-12.45**	0.31	472.00	-40.55	< **2e-16**

Table C.9: Secondary regression summary table for *zero-shot* setting. Significant coefficients and p values in bold. This regression covers all values of alpha {0.1, 0.2, 0.3, 0.4}, which are only tested in experiments with 100k training steps. Thus, the `lapt_steps` variable is excluded from this regression. Formula:
`lmer(accuracy ~ vocab_size + lapt_alpha + task + (1 | language))`