Targeted Multilingual Adaptation for Low-resource Language Families

C.M. Downey$^{\alpha\beta}$ **Terra Blevins**$^{\gamma}$
Dhwani Serai$^{\delta}$ **Dwija Parikh**$^{\delta}$ **Shane Steinert-Threlkeld**$^{\delta}$
$^{\alpha}$Department of Linguistics, University of Rochester
$^{\beta}$Goergen Institute for Data Science, University of Rochester
$^{\gamma}$Paul G. Allen School of Computer Science & Engineering, University of Washington
$^{\delta}$Department of Linguistics, University of Washington
correspondence: c.m.downey@rochester.edu, blvns@cs.washington.edu

Abstract

Massively multilingual models are known to have limited utility in any one language, and to perform particularly poorly on low-resource languages. By contrast, targeted multinguality has been shown to benefit low-resource languages. To test this approach more rigorously, we systematically study best practices for adapting a pre-trained model to a language family. Focusing on the Uralic family as a test case, we adapt XLM-R under various configurations to model 15 languages; we then evaluate the performance of each experimental setting on two downstream tasks and 11 evaluation languages. Our adapted models significantly outperform mono- and multilingual baselines. A regression analysis reveals that adapted vocabulary size is relatively unimportant for low-resource languages, and that low-resource languages can be aggressively up-sampled during training at little detriment to performance in high-resource languages. These results introduce new best practices for performing language adaptation in a targeted setting.

1 Introduction

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). This paper contributes to 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) by analyzing what factors matter most for this approach.

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 languages like Sámi and Erzya, which are extremely data-scarce. Our primary adaptation techniques 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.

We demonstrate that adaptation to a language family is both more efficient than training monolingual models, and performs as well or better on downstream tasks. We also conduct a regression analysis to assess the impact of Lapt steps, adapted vocabulary size, and language sampling alpha in order to recommend *best practices* for adapting models to targeted multilingual groupss. 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.

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 statistical analysis of important parameters for multilingual adaptation to test their relative effects on downstream task performance. 3) We use this analysis to make best-practice recommendations 4) 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.

Findings of the Association for Computational Linguistics: EACL 2024, pages 15647–15663
November 12-16, 2024 ©2024 Association for Computational Linguistics

2 Related Work

Pre-trained model adaptation Extensive work has proposed reusing pre-trained models for new settings to retain existing model knowledge and reduce training costs. Gururangan et al. (2020) show that continued training on domain-specific data effectively adapts models 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 to Uralic languages specifically. They find that vocabulary overlap (and script) is extremely important for transfer success, rather than the language-relatedness of the source model.

Model vocabulary and script A major limitation to adapting models to new languages is the vocabulary, which often fails to cover unseen scripts (Pfeiffer et al., 2021) or tokenizes target text inefficiently (Ács, 2019; Ahia et al., 2023). However, Muller et al. (2021) demonstrate that script is a critical factor in predicting transfer success.

Many adaptation techniques have been proposed to overcome this issue, such as extending the vocabulary with new tokens (Chau et al., 2020; Wang et al., 2020a; Liang et al., 2023) or retraining the vocabulary and embeddings from scratch (Artetxe et al., 2020; de Vries and Nissim, 2021). Other work reuses information in pre-trained embeddings rather than randomly initializing new ones. These include scaling up embedding spaces from smaller target language models (de Vries and Nissim, 2021; Ostendorff and Rehm, 2023) or copying embeddings from the original vocabulary where there is exact overlap (Pfeiffer et al., 2021).

We follow recent works that re-initialize the vocabulary (and embeddings) based on the structure of the original model's embedding space (Minixhofer et al., 2022; Ostendorff and Rehm, 2023; Liu et al., 2024, i.a.). Dobler and de Melo (2023) introduce the FOCUS algorithm, which initializes new token embeddings as a linear combination of the old embeddings for the most semantically similar tokens, as computed by an auxiliary embedding model. Alternatively, Downey et al. (2023) propose heuristics for initializing a new embedding matrix based on script-wise distributions, which perform similarly to more complex techniques like FOCUS.

Targeted multilingualism Recent work has proposed *targeted* or *linguistically-informed* multilingual models instead of the "massively-multilingual" approach covering as many languages as feasible. 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. have similar word order).

Ogueji et al. (2021) train a multilingual model from scratch on 11 African languages, obtaining performance as good or better than XLM-R. Ogunremi et al. (2023) refine this approach by showing that training on languages from individual African language families is more data-efficient than using a mixture of unrelated African languages. Snæbjarnarson et al. (2023) take a similar approach by adapting a Germanic model to Faroese.

Other work uses targeted multilingual training as an *adaptation* process of a pre-trained model, rather than training from scratch. Alabi et al. (2022) adapt XLM-R to 17 African languages via LAPT while discarding the XLM-R vocabulary unused by the target languages. Ljubešić et al. (2024) similarly use LAPT to adapt XLM-R to the closely related languages of Bosnian, Croatian, Montenegrin, and Serbian; and Senel et al. (2024) adapt XLM-R separately to five low-resource Turkic languages while showing that including the high-resource Turkish language during training improves this adaptation.

Our 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.

3 Experiments

We adapt models using Language-Adaptive Pre-Training (LAPT, Chau et al., 2020) and vocabulary specialization (Downey et al., 2023, i.a.) on Uralic language data and compare to both multilingual and monolingual baselines. Within our multilingual experiments, we explicitly model the influence of important hyper-parameters on downstream performance using a linear mixed-effects regression.

Languages First, we obtain raw-text LAPT data for as many Uralic languages as possible. For the high-resource languages (Estonian, Finnish, Hungarian, and Russian), all training data is sourced from the multilingual OSCAR corpus v.22.01 (Abadji et al., 2022), which also contains a small amount of 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).

Table 1 shows an inventory of the LAPT text with total data amounts after combining the corpora for each language. We cover 6/8 Uralic branches, lacking only Ob-Ugric and Samoyedic (Austerlitz, 2008). The resource gap between high- and low-resource languages is stark: Estonian (fourth highest-resource) has ∼1000x more data than the next highest (Komi). The four highest-resource languages were also seen during XLM-R's training, while the rest were not. We treat this as the cutoff between the "high-resource" and "low-resource" Uralic languages for the remainder of this work.

We also 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 using 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 performance, since script overlap is 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 and an official language throughout Russian territory (Austerlitz, 2008).

During training, we sample languages according to a multinomial distribution parameterized by the hyperparameter α (Conneau and Lample, 2019; Conneau et al., 2020a, i.a.; see Figure 1). Languages are sampled by sentence, allowing multiple languages to be potentially 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] testing vocabulary sizes of 16k, 32k, and 64k tokens.[2] We train multilingual tokenizers with a consistent sampling parameter of $\alpha = 0.2$.[3] We re-initialize the model's embedding matrix for the new vocabulary using the FOCUS algorithm (Dobler and de Melo, 2023).

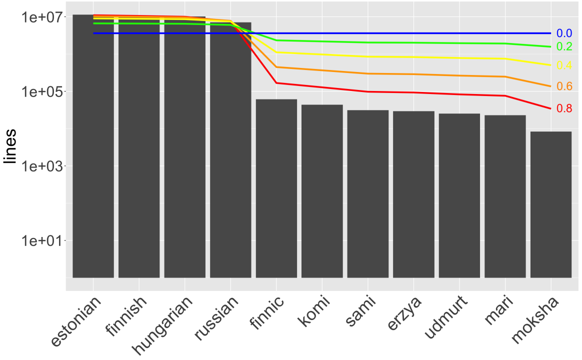

Figure 1: Uralic data composition by number of lines, on a log scale. The actual data quantities are shown with bars, while sampling distributions with several values of the α parameter are plotted as lines

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. When training a new vocabulary, the transformer blocks are frozen for the first

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

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

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

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 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.

10k steps, to prevent model overfitting on the initial (possibly poor) embedding initializations.

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 1), we cap the four high-resource languages at approximately 2 GB of text each.[4] Because several languages of 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 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) on Universal Dependencies (UD) treebanks (de Marneffe et al., 2021).[5] Treebanks exist for all high-resource languages plus Erzya, North Sámi, Komi, Karelian, Livvi, Moksha, and Skolt Sámi. Models are fine-tuned for each task over four random seeds.

Because the amount of fine-tuning data varies considerably over languages, we consider three evaluation settings: *few-shot*, *full-finetune*, and *zero-shot*. For *few-shot*, models are fine-tuned on 512 sampled sentences per language. For *full-finetune*, models are fine-tuned on all examples in a given language (ranging from 896 sentences for Erzya to 32,768 for Russian). *Zero-shot* setting is used for very low-resource languages which have only small test sets, and no standard training data: we fine-tune the model on the combination of languages with training sets, and then evaluate directly on the target test set. An inventory of Uralic UD data can be found in Appendix B, 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) without modification. We also test XLM-R adapted with LAPT, but without vocabulary specialization. LAPT alone is a strong baseline, but as Downey et al. (2023) note, training a large cross-lingual vocabulary incurs considerable extra computation compared to a smaller, specialized one. Given cross-lingual tokenizers are often inefficient and ineffective for low-resource languages (Ács, 2019; Rust et al., 2021), we hypothesize a specialized vocabulary will show a performance advantage as well as reduced computational cost.

We also train baselines adapted to single languages, adapting XLM-R with LAPT and a vocab size of 16k (per language). We assume a shared computational "budget" of 400k training steps, where steps are allocated across languages according to the multinomial distribution with $\alpha = 0.1$, similar to the data sampling technique for multilingual training. This baseline is meant to be comparable to our multilingual model trained with 400k steps, vocab size 16k, and $\alpha = 0.1$.

[4] This is in addition to alpha sampling, reflected in Figure 1.

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

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 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 § 3.

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 3: Zero-shot comparisons with monolingual baselines (both tasks) with the same models as Table 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.

4 Results

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

4.1 Baselines

Monolingual baselines Tables 2 and 3 compare our best-performing, fully-adapted multilingual models to the comparable monolingual baselines described in §3. With a few 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 3), and low-resource languages.

Multilingual baselines Tables 4 and 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 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 shows that in *few-shot* evaluations, our smallest model with vocabulary specialization significantly outperforms the best baseline model without. An adapted vocabulary of 16k tokens results in an average gain of 1.6 points over the baseline, and increasing to 64k tokens raises this to 4.7 points. We also note that LAPT on XLM-R with its original vocabulary incurs ∼2-3x more computation than a version with a specialized vocabulary of size 32k.

The *zero-shot* evaluations do not reflect this consistent improvement with increasing vocabulary size (Table 5). 4/5 zero-shot languages still see the best results with a specialized vocabulary. The exception is Skolt Sámi, which is modeled best by the +LAPT/-vocab-adaptation baseline. However, our results for Skolt Sámi go against overall trends in our experiments, and we delve into this finding further with an error analysis in Appendix D.

For space and clarity, we focus only on UAS results in this section. Comparable tables for POS are found in Appendix C. We observe similar trends for POS, though the LAPT baseline with original vocabulary is more on par with the specialized vocabularies. We hypothesize that this reflects POS being a simpler task than dependency parsing, since

[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.

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: Few-shot UAS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocab specialization). Second row is XLM-R with original vocabulary, but adapted with Uralic LAPT

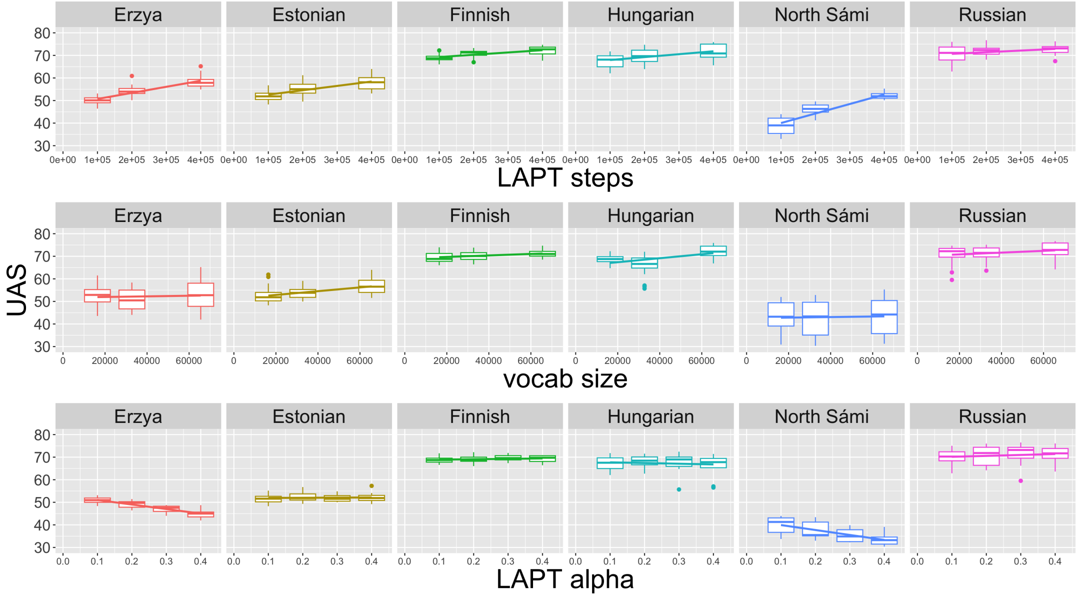

Figure 2: Few-shot UAS — effect of hyper-parameters by language, marginalized across other parameter settings

the latter involves more linguistic structure. It may be telling that the advantage of adapted vocabularies is clearer in the more complicated UAS task.

4.2 Qualitative trends

Figure 2 shows the effect of each hyper-parameter (marginalized across other parameters) in the *few-shot* setting for UAS.[7] Unsurprisingly, the number of LAPT (training) steps has a large effect on performance. This also reflects that adaptation may take a long time to properly converge on new languages, as the slope is steeper for languages new to XLM-R such as Erzya (myv). Adapted vocabulary size also has an overall positive effect on performance, though this effect is not as strong as longer training and not as clear for the low-resource Erzya (myv) and North Sámi (sme). Finally, the effect of sampling alpha diverges between high- and low-resource languages, as lower alpha values upsample low-resource languages and down-sample high-resource ones. However, the observed effect for low-resource languages at lower alpha values is much larger than the corresponding degradation on high-resource languages.

Plots for the *zero-shot* setting. The effects of steps and alpha are similar to the *few-shot* trends. However, vocabulary size does not have an obvious effect in this setting (an observation corroborated by our regression). As previously noted, Skolt Sámi performance remains consistently poor across hyperparameters (see § D).

4.3 Statistical analysis

Experimental Setup We conduct a regression analysis to predict performance with linear mixed-effect models in the lme4 package (Bates et al., 2015). LAPT steps and vocabulary size are treated as fixed continuous effects. Fine-tuning examples is treated as a fixed continuous effect for *few-shot* and *full-finetune* settings only. Task (POS vs UAS) is modeled as a fixed categorical effect, as the scores mostly follow a fixed offset (with POS ac-

[7] We see similar trends in our regression analysis for the POS tagging task (Appendix Figure 4).

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 5: Zero-shot UAS — comparison with multilingual baselines. First row is XLM-R "off-the-shelf" (without LAPT or vocab specialization). Second row is XLM-R with original vocabulary, but adapted with Uralic LAPT

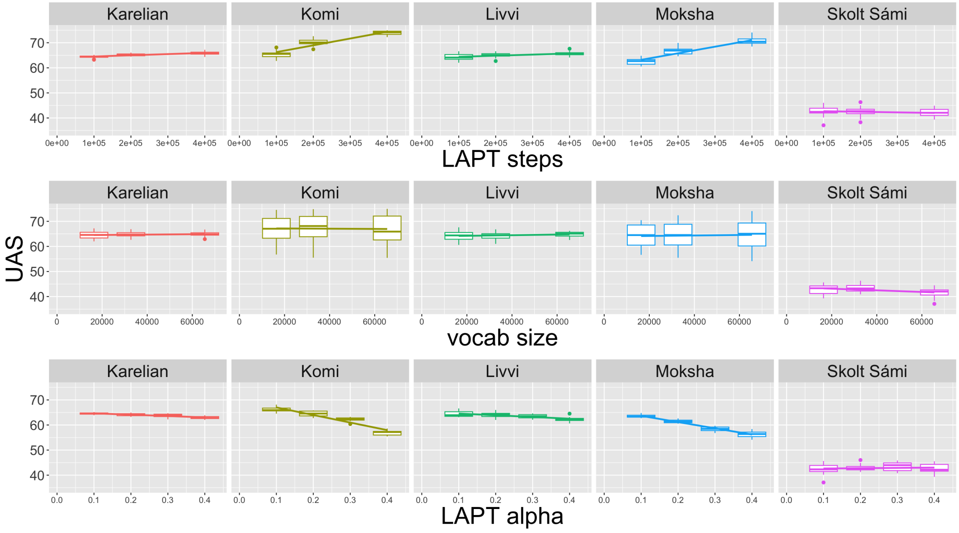

Figure 3: Zero-shot UAS — effect of hyper-parameters by language, marginalized across other parameter settings

curacy higher than UAS). We verify this with a regression with interaction terms between task and other parameters (e.g. steps), finding no significant interactions; ANOVA confirms no significant difference between the regressions ($p = 0.95$).

Since the effect of α has a different sign and magnitude between high- and low-resource languages, we model it as an *interaction* with a binary variable coding the language as high- or low-resource. We justify the binarity by the stark resource disparity between the two groups (§ 3). Due to differing baseline performance across languages, we also include a language-wise random-effect intercept. The final regression formula and full summary table with coefficients are in Appendix Table 12.

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.[8] Sampling alpha is significant in the low-resource case ($p = 0.035$), but not the high-resource ($p = 0.36$). This indicates lower alpha has a significant positive effect for low-resource languages, 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 concrete terms, this means doubling steps from 100k to 200k is ∼2.7 times as effective as doubling the vocabulary from 16k to 32k tokens, and ∼4.2 times as effective as doubling the fine-tuning examples to 1024. Finally, we test for but find no significant interaction between steps and vocabulary size; ANOVA comparison confirms no significant difference between models with and without this interaction ($p = 0.43$).

Zero-shot Results Our *zero-shot* regression is similar to the previous, except that number of fine-tuning examples is not applicable, and there is no interaction between alpha and resource level, since all zero-shot languages are low-resource. The effects for steps and task are highly significant ($p < 0.001$); alpha is also significant ($p = 0.0027$).

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

In contrast to the fine-tuned settings, vocabulary size is not significant ($p = 0.73$). The coefficient for steps is 1.35 points per 100k steps, which is slightly smaller than for the fine-tuned experiments. This could be partly due to the results for Skolt Sámi showing little change under any parameters.

5 Discussion

Multilinguality is beneficial for many languages The baselines in §4.1 demonstrate that, given a fixed computational budget, it is more effective to adapt a multilingual model to cover a group of related languages than to adapt models monolingually. This is especially true for low-resource languages, but surprisingly some high-resource ones (like Hungarian and Russian) also benefit from multilinguality. This supports the idea that multilingual training is useful for learning general patterns that are beneficial for many languages. Table 3 further shows that robust performance on languages without fine-tuning data, like Komi and Moksha, is only feasible by combining multilinguality and transfer learning.

Specialized vocab is more effective and efficient Our multilingual baselines in §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 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.[9]

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 6: Total model parameters and mean sequence length for each vocabulary size. In parentheses is percent change from the next-smallest vocab. Computed on 100k sentences sampled from LAPT data ($\alpha = 0.1$).

Training steps vs. vocabulary size Though we find that training steps and vocabulary size both positively contribute to downstream performance in fine-tuned settings, an additional 100k steps is almost three times as effective as adding 16k additional tokens (§4.3). These factors also come with efficiency tradeoffs: 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), whereas doubling the training steps doubles the number of operations.

Furthermore, while a larger vocabulary size can reduce sequence length as the SentencePiece model becomes more efficient, each doubling of the vocabulary size only reduces the average sequence length about 10% (Table 6). These parameter increases therefore eventually outpace efficiency from shorter sequences, as well as increase the model's memory footprint and hardware costs.

Finally, our regression shows that vocabulary size does not significantly affect *zero-shot* task performance, which covers our lowest-resource languages (§4.3 and Table 5). Therefore, a best practice for adapting to a low-resource language family would 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 should then be spent on longer training rather than a larger model.

Lower alpha is better While α does not have a significant effect on task performance in high-resource languages, low alphas *do* significantly benefit low-resource settings (§4.3). Our multilingual models thus frequently achieve their best average performance with the lower $\alpha = 0.1$, buoyed by strong performance in low-resource languages.

This indicates that practitioners can aggressively up-sample low-resource languages in multilingual datasets with little risk to performance on high-resource "anchor" languages. Further, as low as $\alpha = 0.1$, we see no evidence that "over-sampling" low-resource languages harms their downstream performance. However, we note that the considered high-resource languages are in XLM-R's original training set, which likely affects the model's robustness on these languages. It is an open question whether multilingual sampling dynamics differ in "from-scratch" training scenarios or for other high-resource, but previously unseen, languages.

[9] 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.

6 Conclusion

This work shows that adapting a pre-trained cross-lingual model to a language family greatly improves task performance, especially for under-resourced languages in that family. Targeted multilingual adaptation soundly outperforms monolingual adaptation 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 the large "cross-lingual" vocabulary of XLM-R, while simultaneously improving computational and memory efficiency. Finally, our statistical analysis demonstrates which parameters (sampling alpha, training steps) are key to targeted adaptation performance.

While our work is similar in goal to recent projects such as Glot500-m (Imani et al., 2023), our approach contrasts in important ways. Imani et al. (2023); Liang et al. (2023); *i.a.* propose "horizontal" scaling to more languages primarily through vocabulary *expansion*, growing the overall model size. We on the other hand advocate for model *specialization*, shrinking the overall model size while increasing per-language parameter and vocabulary capacity. Both our work and Glot500-m significantly outperform XLM-R on under-resourced languages (though the exact evaluation results are not directly comparable[10]). However, we note that our smallest fine-tuned model does so with just 25% of the parameters of Glot500-m.

Specialization to smaller models has significant advantages in resource-constrained scenarios, and may better facilitate the democratization of NLP research and engineering for more language communities. Imani et al. (2023) point out that reducing the size of models like Glot500-m through knowledge distillation would be useful future work. In fact, the two approaches are not mutually exclusive: our specialization technique could be applied to larger base models with expanded multilinguality, and model *expansion* and *specialization* may be more or less practical depending on the specific applied NLP workflow.

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, as it both leverages the benefit of multilingualism for under-resourced languages and avoids the "Curse of Multilinguality" seen in massively multilingual approaches. However, given the success of large pre-trained language models, and of the pre-training paradigm more generally (Gururangan et al., 2020), we argue that it is more effective to leverage transferable information in existing cross-lingual models, rather than training from scratch. We hope our findings will inform best practices for such targeted multilingual adaption when extending the benefits of pre-trained models to under-resourced languages.

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.

[10] Both studies evaluate POS tagging on Erzya, North Sámi, Estonian, Finnish, Hungarian, and Russian. However, we do so in a fine-tuned setting whereas Glot500-m is fine-tuned on English then zero-shot transferred to all other languages. Our models show higher accuracy on low-resource languages, but this may be attributable to the difference in setting among other confounding variables like total compute budget for continued pre-training.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Judit Ács. 2019. Exploring BERT's Vocabulary.

Judit Ács, Dániel Lévai, and Andras Kornai. 2021. Evaluating transferability of BERT models on Uralic languages. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*, pages 8–17, Syktyvkar, Russia (Online). Association for Computational Linguistics.

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

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pretrained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Robert Austerlitz. 2008. Uralic languages. In Bernard Comrie, editor, *The World's Major Languages*, 3 edition, pages 477–483. Routledge, London, UK.

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

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When Is Multilinguality a Curse? Language Modeling for 250 High- and Low-Resource Languages. *arXiv preprint*. ArXiv:2311.09205 [cs].

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

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32, Vancouver, Canada. Curran Associates, Inc.

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

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Wietse de Vries and Malvina Nissim. 2021. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online. Association for Computational Linguistics.

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

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

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *International Conference on Learning Representations*.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 434–452, Online only. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2024. *Speech and Language Processing*, 3 edition.

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

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

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. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

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

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

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

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schuetze. 2024. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1067–1097, Mexico City, Mexico. Association for Computational Linguistics.

Nikola Ljubešić, Vít Suchomel, Peter Rupnik, Taja Kuzman, and Rik van Noord. 2024. Language Models on a Diet: Cost-Efficient Development of Encoders for Closely-Related Languages via Additional Pretraining. *arXiv preprint*. ArXiv:2404.05428 [cs].

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *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. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

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

Tolulope Ogunremi, Dan Jurafsky, and Christopher Manning. 2023. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1251–1266, Dubrovnik, Croatia. Association for Computational Linguistics.

Malte Ostendorff and Georg Rehm. 2023. Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning. *arXiv preprint*. ArXiv:2301.09626 [cs].

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Lütfi Kerem Senel, Benedikt Ebing, Konul Baghirova, Hinrich Schuetze, and Goran Glavaš. 2024. Kardeş-NLU: Transfer to low-resource languages with the help of a high-resource cousin – a benchmark and evaluation for Turkic languages. In *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. Association for Computational Linguistics.

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

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020a. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

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

A Training Details

The main details of our experimental process can be found in § 3. Here we provide our choice of hyperparameters and other details relevant to reproducibility.

A.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.

A.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 7.

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 7: Hyperparameters for model training (LAPT)

B Evaluation Details

B.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. 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 8.

B.2 Parameters

Hyperparameters for task fine-tuning on POS and UAS are in Table 9. We cap fine-tuning training data at 32,768 sequences (only relevant for Russian).

B.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 → j, is then assigned according to a biaffine scoring function:

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

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

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

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

where U, W, and b are weights learned by the model. A probability distribution over possible heads is then computed by passing score(i → 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 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 10, 11). This is possibly due to the relative simplicity of the task. The parameter-wise trends for POS are mostly the same as for UAS (Figures 4, 5).

D 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.[11] Single-character tokens like <õ>, <ä>, <â>, and <å> also

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

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 8: Universal Dependencies evaluation set sizes, by number of examples (sentences)

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 9: 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

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 UNK tokens and facilitate better transfer learning between the language-modeling and task domains. Particular OOV tokens included `<_de>`, `<_di>`, `<son>`, `<_â'tte>`.

E Regression tables

The full regression summaries from the `lme4` package (Bates et al., 2015) can be found in Tables 12-15. These cover both the fine-tuned (*few-shot*/*full-finetune*) and *zero-shot* models. As mentioned in § 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 13 and 15). 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 10: 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 11: 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

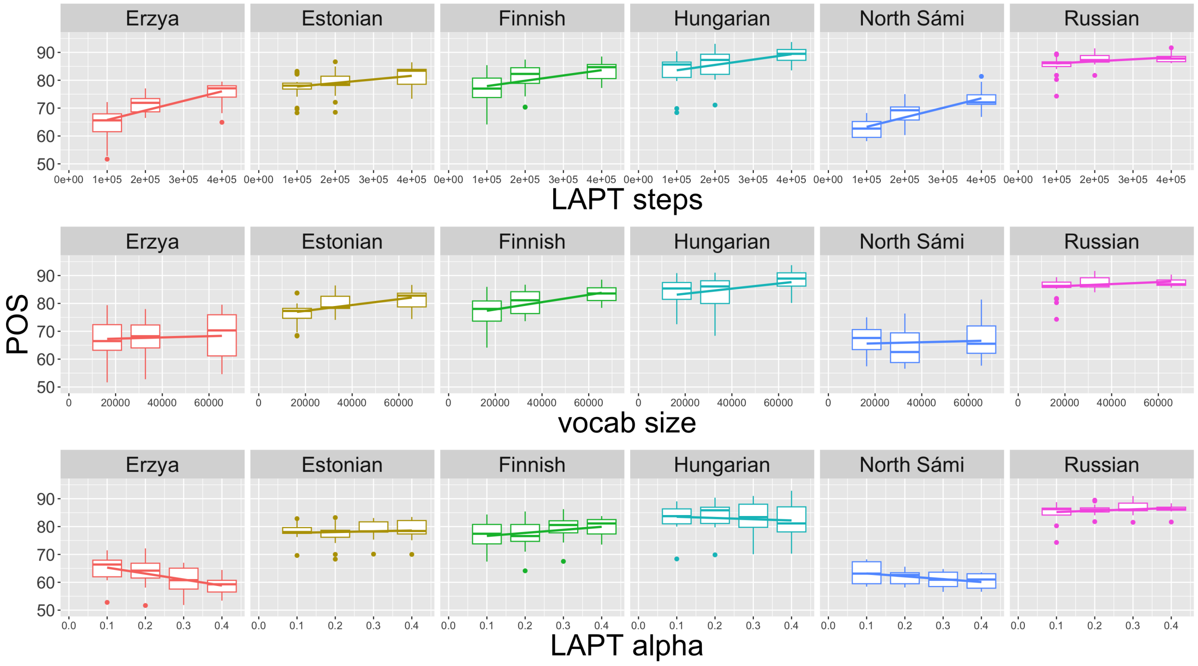

Figure 4: Few-shot POS — effect of hyper-parameters by language, marginalized across other parameter settings

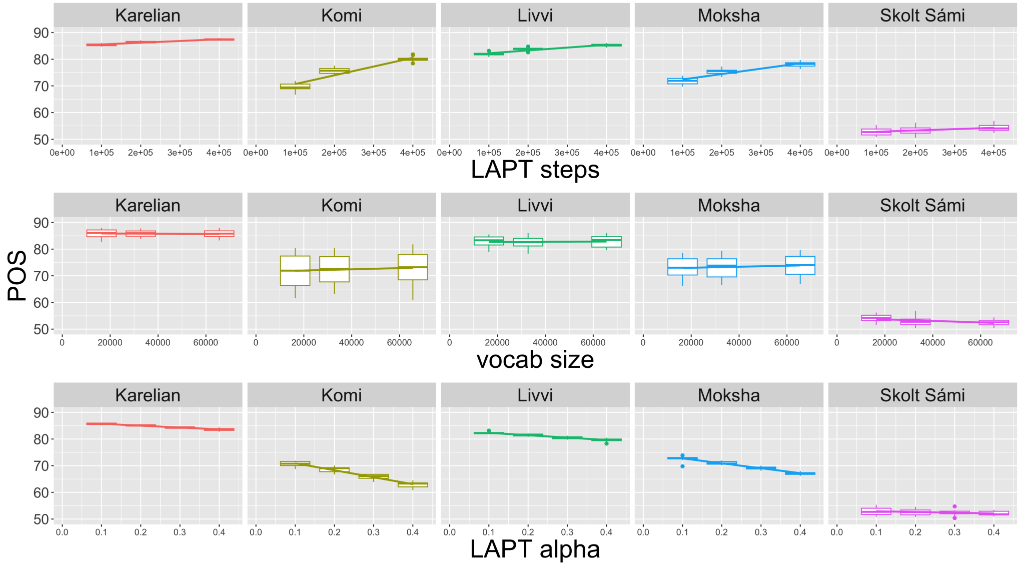

Figure 5: Zero-shot POS — effect of hyper-parameters by language, marginalized across other parameter settings

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 12: 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 13: 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 14: 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 15: 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))`