Specializing Multilingual Language Models: An Empirical Study

Ethan C. Chau[†] **Noah A. Smith**[†⋆]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[⋆]Allen Institute for Artificial Intelligence
{echau18,nasmith}@cs.washington.edu

Abstract

Pretrained multilingual language models have become a common tool in transferring NLP capabilities to low-resource languages, often with adaptations. In this work, we study the performance, extensibility, and interaction of two such adaptations: vocabulary augmentation and script transliteration. Our evaluations on part-of-speech tagging, universal dependency parsing, and named entity recognition in nine diverse low-resource languages uphold the viability of these approaches while raising new questions around how to optimally adapt multilingual models to low-resource settings.

1 Introduction

Research in natural language processing is increasingly carried out in languages beyond English. This includes *high-resource* languages with abundant data, as well as *low-resource* languages, for which labeled (and unlabeled) data is scarce. In fact, many of the world's languages fall into the latter category, even some with a high number of speakers. This presents unique challenges compared to high-resource languages: effectively modeling low-resource languages involves both accurately tokenizing text in such languages and maximally leveraging the limited available data.

One common approach to low-resource NLP is the *multilingual* paradigm, in which methods that have shown success in English are applied to the union of many languages' data,[1] enabling transfer between languages. For instance, multilingual contextual word representations (CWRs) from language models (Devlin et al., 2019; Huang et al., 2019; Lample and Conneau, 2019, *inter alia*) are conventionally "pretrained" on large multilingual corpora before being "finetuned" directly on supervised tasks; this pretraining-finetuning approach is derived from analogous monolingual models (Devlin et al., 2019; Liu et al., 2019; Peters et al., 2018). However, considering the diversity of the world's languages and the great data imbalance among them, it is natural to question whether the current multilingual paradigm can be improved upon for low-resource languages.

Indeed, past work has demonstrated that it can. For instance, Wu and Dredze (2020) find that multilingual models often lag behind non-contextualized baselines for the lowest-resource languages in their training data, drawing into question their utility in such settings. Conneau et al. (2020a) posit that this phenomenon is a result of limited model capacity, which proves to be a bottleneck for sufficient transfer to low-resource languages. In fact, with multilingual models only being pretrained on a limited set of languages, most of the world's languages are unseen by the model. For such languages, the performance of such models is even worse (Chau et al., 2020), due in part to the diversity of scripts across the world's languages (Muller et al., 2021; Pfeiffer et al., 2021b; Rust et al., 2021) as compared to the models' Latin-centricity (Ács, 2019).

Nonetheless, there have been multiple attempts to remedy this discrepancy by *specializing*[2] a multilingual model to a given target low-resource language, from which we take inspiration. Among them, Chau et al. (2020) augment the model's vocabulary to more effectively tokenize text, then pretrain on a small amount of data in the target language; they report significant performance improvements on a small set of low-resource languages. In a similar vein, Muller et al. (2021) propose to transliterate text in the target language

[1]Within the multilingual paradigm, a distinction is sometimes made between *massively multilingual* methods, which consider tens or hundreds of languages; and *polyglot* methods, which use only a handful. In this paper, all mentions of "multilingual" refer to the former.

[2]We use *specialization* to denote preparing a model for use on a specific target language, to the exclusion of others. This is a subset of *adaptation*, which includes all techniques that adjust a model for use on target languages, regardless of their resulting universality.

to Latin script to be better tokenized by the existing model, followed by additional pretraining; they observe mixed results and note that transliteration quality may be a confounding factor. We hypothesize that these two methods can serve as the basis for improvements in modeling a broad set of low-resource languages.

In this work, we study the effectiveness, extensibility, and interaction of these two approaches to specialization: the vocabulary augmentation technique of Chau et al. (2020) and the script transliteration method of Muller et al. (2021). We verify the performance of vocabulary augmentation on three tasks in a diverse set of nine low-resource languages across three different scripts, especially on non-Latin scripts (§2) and find that these gains are associated with improved vocabulary coverage of the target language. We further observe a negative interaction between vocabulary augmentation and transliteration in light of a broader framework for specializing multilingual models, while noting that vocabulary augmentation offers an appealing balance of performance and cost (§3). Overall, our results highlight several possible directions for future study in the low-resource setting. Our code, data, and hyperparameters are publicly available.[3]

2 Revisiting Vocabulary Augmentation

We begin by revisiting the Vocabulary Augmentation method of Chau et al. (2020), which we recast more generally in light of recent work (§2.1). We evaluate their claims on three different tasks, using a diverse set of languages in multiple scripts (§2.2), and find that the results hold to an even more pronounced degree in unseen low-resource languages with non-Latin scripts (§2.3).

2.1 Method Overview

Following Chau et al. (2020), we consider how to apply the pretrained multilingual BERT model (MBERT; Devlin et al., 2019) to a target low-resource language, for which both labeled and unlabeled data is scarce. This model has produced strong CWRs for many languages (Kondratyuk and Straka, 2019, *inter alia*) and has been the starting model for many studies on low-resource languages (Muller et al., 2021; Pfeiffer et al., 2020; Wang et al., 2020). MBERT covers the languages with the 104 largest Wikipedias, and it uses this data to construct a wordpiece vocabulary (Wu et al., 2016) and train its transformer-based architecture (Vaswani et al., 2017). Although low-resource languages are slightly oversampled, high-resource languages still dominate both the final pretraining data and the vocabulary (Ács, 2019; Devlin et al., 2019).

Chau et al. (2020) note that target low-resource languages fall into three categories with respect to MBERT's pretraining data: the lowest-resource languages in the data (Type 1), completely unseen low-resource languages (Type 2), and low-resouce languages with more representation (Type 0).[4] Due to their poor representation in the vocabulary, Type 1 and Type 2 languages achieve suboptimal tokenization and higher rates of the "unknown" wordpiece[5] when using MBERT out of the box. This hinders the model's ability to capture meaningful patterns in the data, resulting in reduced data efficiency and degraded performance.

We note that this challenge is exacerbated when modeling languages written in non-Latin scripts. MBERT's vocabulary is heavily Latin-centric (Ács, 2019; Muller et al., 2021), resulting in a significantly larger portion of non-Latin scripts being represented with "unknown" tokens (Pfeiffer et al., 2021b) and further limiting the model's ability to generalize. In effect, MBERT's low initial performance on such languages can be attributed to its inability to represent the script itself.

To alleviate the problem of poor tokenization, Chau et al. (2020) propose to specialize MBERT using Vocabulary Augmentation (VA). Given unlabeled data in the target language, they train a new wordpiece vocabulary on the data, then select the 99 most common wordpieces in the new vocabulary that replace "unknown" tokens under the original vocabulary. They then add these 99 wordpieces to the original vocabulary and continue pretraining MBERT on the unlabeled data for additional steps. They further describe a tiered variant (TVA), in which a larger learning rate is used for the embeddings of these 99 new wordpieces. VA yields strong gains over unadapted multilingual language models on dependency parsing in four low-resource languages with Latin scripts. How-

[3] https://github.com/ethch18/specializing-multilingual

[4] Muller et al. (2021) further subdivide Type 2 into Easy, Medium, and Hard languages, based on the performance of MBERT after being exposed to these languages. However, this categorization cannot be determined *a priori* for a given language.

[5] The "unknown" wordpiece is inserted when the wordpiece algorithm is unable to segment a word-level token with the current vocabulary.

ever, no evaluation has been performed on other tasks or on languages with non-Latin scripts, which raises our first research question:

RQ1: Do the conclusions of Chau et al. (2020) hold for other tasks and for languages with non-Latin scripts?

We can view VA and TVA as an instantation of a more general framework of vocabulary augmentation, shared by other approaches to using MBERT in low-resource settings. Given a new vocabulary V, number of wordpieces n, and learning rate multiplier a, the n most common wordpieces in V are added to the original vocabulary. Additional pretraining is then performed, with the embeddings of the n wordpieces taking on a learning rate a times greater than the overall learning rate. For VA, we set $n = 99$ and $a = 1$, while we treat a as a hyperparameter for TVA. The related E-MBERT method of Wang et al. (2020) sets $n = |V|$ and $a = 1$. Investigating various other instantiations of this framework is an interesting research direction, though it is out of the scope of this work.

2.2 Experiments

We expand on the dependency parsing evaluations of Chau et al. (2020) by additionally considering named entity recognition and part-of-speech tagging. We follow Kondratyuk and Straka (2019) and compute the CWR for each token as a weighted sum of the activations at each MBERT layer. For dependency parsing, we follow the setup of Chau et al. (2020) and Muller et al. (2021) and use the CWRs as input to the graph-based dependency parser of Dozat and Manning (2017). For named entity recognition, the CWRs are used as input to a CRF layer, while part-of-speech tagging uses a linear projection atop the representations. In all cases, the underlying CWRs are finetuned during downstream task training, and we do not add an additional encoder layer above the transformer outputs. We train models on five different random seeds and report average scores and standard errors.

2.2.1 Languages and Datasets

We select a set of nine typologically diverse low-resource languages for evaluation, including three of the original four used by Chau et al. (2020). These languages use three different scripts and are chosen based on the availability of labeled datasets and their exemplification of the three language types identified by Chau et al. (2020). Of the languages seen by MBERT, all selected Type 0 languages are within the 45 largest Wikipedias, while the remaining Type 1 languages are within the top 100. The Type 2 languages, which are excluded from MBERT, are all outside of the top 150.[6] Additional information about the evaluation languages is given in Tab. 1.

Unlabeled Datasets Following Chau et al. (2020), we use articles from Wikipedia as unlabeled data for additional pretraining in order to reflect the original pretraining data. We downsample full articles from the largest Wikipedias to be on the order of millions of tokens in order to simulate a low-resource unlabeled setting, and we remove sentences that appear in the labeled validation or test sets.

Labeled Datasets For dependency parsing and part-of-speech tagging, we use datasets and train/test splits from Universal Dependencies (Nivre et al., 2020), version 2.5 (Zeman et al., 2019). POS tagging uses language-specific part-of-speech tags (XPOS) to evaluate understanding of language-specific syntactic phenomena. The Belarusian treebank lacks XPOS tags for certain examples, so we use universal part-of-speech tags instead. Dependency parsers are trained with gold word segmentation and no part-of-speech features. Experiments with named entity recognition use the WikiAnn dataset (Pan et al., 2017), following past work (Muller et al., 2021; Pfeiffer et al., 2020; Wu and Dredze, 2020). Specifically, we use the balanced train/test splits of (Rahimi et al., 2019). We note that UD datasets were unavailable for Meadow Mari, and partitioned WikiAnn datasets were missing for Wolof.

2.2.2 Baselines

To measure the effectiveness of VA, we benchmark it against unadapted MBERT, as well as directly pretraining MBERT on the unlabeled data without modifying the vocabulary (Chau et al., 2020; Muller et al., 2021; Pfeiffer et al., 2020). Following Chau et al. (2020), we refer to the latter approach as *language-adaptive pretraining* (LAPT). We also evaluate two monolingual baselines that are trained on our unlabeled data: fastText embeddings (FASTT; Bojanowski et al., 2017), which represent a static word vector approach; and a BERT model trained from scratch (BERT). For

[6] Based on https://meta.wikimedia.org/wiki/List_of_Wikipedias.

Language	Type	Script	Family	# Sentences	# Tokens	Downsample %	# WP/Token
Bulgarian (BG)	0	Cyrillic	Slavic	357k	5.6M	10%	1.81
Belarusian (BE)	0	Cyrillic	Slavic	187k	2.7M	10%	2.25
Meadow Mari (MHR)	2	Cyrillic	Uralic	52k	512k	–	2.37
Vietnamese (VI)	0	Latin	Viet-Muong	338k	6.9M	5%	1.17
Irish (GA)	1	Latin	Celtic	274k	5.8M	–	1.83
Maltese (MT)	2	Latin	Semitic	75k	1.4M	–	2.39
Wolof (WO)	2	Latin	Niger-Congo	15k	396k	–	1.78
Urdu (UR)	0	Perso-Arabic	Indic	201k	3.6M	20%	1.58
Uyghur (UG)	2	Perso-Arabic	Turkic	136k	2.3M	–	2.54

Table 1: Language overview and unlabeled dataset statistics: number of sentences, number of tokens, and average wordpieces per token under the original MBERT vocabulary.

BERT, we follow Muller et al. (2021) and train a six-layer RoBERTa model (Liu et al., 2019) with a language-specific SentencePiece tokenizer (Kudo and Richardson, 2018). For a fair comparison to VA, we use the same task-specific architectures and modify only the input representations.

2.2.3 Implementation Details

To pretrain LAPT and VA models, we use the code of Chau et al. (2020), who modify the pretraining code of Devlin et al. (2019) to only use the masked language modeling (MLM) loss. To generate VA vocabularies, we train a new vocabulary of size 5000 and select the 99 wordpieces that replace the most unknown tokens. We train with a fixed linear warmup of 1000 steps. To pretrain BERT models, we use the HuggingFace Transformers library (Wolf et al., 2020). Following Muller et al. (2021), we train a half-sized RoBERTa model with six layers and 12 attention heads. We use a byte-pair vocabulary of size 52000 and a linear warmup of 1 epoch. For LAPT, VA, and BERT, we train for up to 20 epochs total, selecting the highest-performing epoch based on validation masked language modeling loss. FASTT models are trained with the skip-gram model for five epochs, with the default hyperparameters of Bojanowski et al. (2017).

Training of downstream parsers and taggers follows Chau et al. (2020) and Kondratyuk and Straka (2019), with an inverse square-root learning rate decay and linear warmup, and layer-wise gradual unfreezing and discriminative finetuning. Models are trained with AllenNLP, version 2.1.0 (Gardner et al., 2018), for up to 200 epochs with early stopping based on validation performance. We choose batch sizes to be the maximum that allows for successful training on one GPU.

2.3 Results

Tab. 2 presents performance of the different input representations on POS tagging, dependency parsing, and named entity recognition. VA achieves strong results across all languages and tasks and is the top performer in the majority of them, suggesting that augmenting the vocabulary addresses MBERT's limited vocabulary coverage of the target language and is beneficial during continued pretraining.

The relative gains that VA provides appear to correlate not only with language type, as in the findings of Chau et al. (2020), but also with each language's script. For instance, in Vietnamese, which is a Type 0 Latin script language, the improvements from VA are marginal at best, reflecting the Latin-dominated pretraining data of MBERT. Irish, the Type 1 Latin script language, is only slightly more receptive. However, Type 0 languages in Cyrillic and Arabic scripts, which are less represented in MBERT's pretraining data, are more receptive to VA, with VA even outperforming all other methods for Urdu. This trend is amplified in the Type 2 languages, as the improvements for Maltese and Wolof are small but significant. However, they are dwarfed in magnitude by those of Uyghur, where VA achieves up to a 57% relative error reduction over LAPT.

This result corroborates the findings of both Chau et al. (2020) and Muller et al. (2021) and answers **RQ1**. Prior to specialization, MBERT is especially poorly equipped to handle unseen low-resource languages and languages in non-Latin scripts due to its inability to model the script itself. In such cases, specialization via VA is beneficial, providing MBERT with explicit signal about the target language and script while maintaining its language-agnostic insights. On the other hand, this also motivates additional investigation into reme-

Rep.	BE* (0)	BG (0)	GA (1)	MT (2)	UG (2)	UR (0)	VI (0)	WO (2)	Avg.
FASTT	68.84 ± 7.16	88.86 ± 0.37	86.87 ± 2.55	89.68 ± 2.15	89.45 ± 1.37	90.81 ± 0.31	81.84 ± 1.15	87.48 ± 0.55	85.48
BERT	91.00 ± 0.30	94.48 ± 0.10	90.36 ± 0.20	92.61 ± 0.10	90.87 ± 0.13	89.88 ± 0.13	84.73 ± 0.13	87.71 ± 0.31	90.20
MBERT	94.57 ± 0.45	96.98 ± 0.08	91.91 ± 0.25	94.01 ± 0.17	78.07 ± 0.22	91.77 ± 0.18	88.97 ± 0.10	93.04 ± 0.20	91.16
LAPT	**95.74 ± 0.44**	97.15 ± 0.04	93.28 ± 0.19	95.76 ± 0.09	79.88 ± 0.27	92.18 ± 0.16	**89.64 ± 0.20**	**94.58 ± 0.13**	92.28
VA	95.28 ± 0.51	**97.20 ± 0.06**	**93.33 ± 0.16**	**96.33 ± 0.09**	**91.49 ± 0.13**	**92.24 ± 0.16**	89.49 ± 0.22	94.48 ± 0.20	**93.73**

(a) POS tagging (accuracy). *Belarusian uses universal POS tags.

Rep.	BE (0)	BG (0)	GA (1)	MT (2)	UG (2)	UR (0)	VI (0)	WO (2)	Avg.
FASTT	35.81 ± 2.24	84.03 ± 0.41	65.58 ± 1.21	68.45 ± 1.40	54.52 ± 1.02	79.33 ± 0.25	54.91 ± 0.79	70.39 ± 1.39	64.13
BERT	45.77 ± 1.35	84.61 ± 0.27	64.02 ± 0.49	65.92 ± 0.45	60.34 ± 0.27	78.07 ± 0.22	54.70 ± 0.27	60.12 ± 0.39	64.19
MBERT	71.83 ± 0.90	91.62 ± 0.23	71.68 ± 0.62	76.63 ± 0.35	47.70 ± 0.44	81.45 ± 0.26	64.58 ± 0.42	76.24 ± 0.83	72.72
LAPT	72.77 ± 1.12	**92.08 ± 0.31**	**74.79 ± 0.12**	81.53 ± 0.37	50.67 ± 0.34	81.78 ± 0.44	**66.15 ± 0.41**	**80.34 ± 0.14**	75.01
VA	**73.22 ± 1.23**	91.90 ± 0.20	74.35 ± 0.22	**82.00 ± 0.31**	**67.55 ± 0.17**	**81.88 ± 0.25**	65.64 ± 0.12	80.22 ± 0.41	**77.09**

(b) UD parsing (LAS).

Rep.	BE (0)	BG (0)	GA (1)	MT (2)	UG (2)	UR (0)	VI (0)	MHR (2)	Avg.
FASTT	84.26 ± 0.86	87.98 ± 0.76	67.21 ± 4.30	33.53 ± 17.89	–	92.85 ± 2.04	85.57 ± 1.98	35.28 ± 13.81	60.84
BERT	88.08 ± 0.62	90.31 ± 0.20	76.58 ± 0.98	54.64 ± 3.51	61.54 ± 3.70	94.04 ± 0.55	88.08 ± 0.15	54.17 ± 2.88	75.93
MBERT	91.13 ± 0.07	92.56 ± 0.09	82.82 ± 0.57	61.86 ± 2.60	50.76 ± 1.86	94.60 ± 0.34	92.13 ± 0.27	61.85 ± 3.25	78.46
LAPT	**91.61 ± 0.74**	**92.96 ± 0.13**	84.13 ± 0.78	**81.53 ± 2.33**	56.76 ± 4.91	95.17 ± 0.29	92.41 ± 0.15	59.17 ± 5.15	81.72
VA	91.38 ± 0.56	92.70 ± 0.11	**84.82 ± 1.00**	80.00 ± 2.77	**68.93 ± 3.30**	**95.43 ± 0.22**	**92.43 ± 0.16**	**64.23 ± 3.07**	**83.74**

(c) NER (macro F1). – indicates that a model did not converge.

Table 2: Results on POS tagging, UD parsing, and NER, with standard deviations from five random initializations. **Bolded** results are the maximum for each language, and scores in gray are not significantly worse than the best model (1-sided paired t-test, $p = 0.05$ with Bonferonni correction).

dies for the script imbalance at a larger scale, e.g., more diverse pretraining data.

2.4 Analysis

We perform further analysis to investigate VA's patterns of success. Concretely, we hypothesize that VA significantly improves the tokenizer's coverage of target languages where it is most successful. Inspired by Ács (2019), Chau et al. (2020), and Rust et al. (2021), we quantify tokenizer coverage using the percentage of tokens in the raw text that yield unknown wordpieces when tokenized with a given vocabulary ("UNK token percentage"). These are tokens whose representations contain at least partial ambiguity due to the inclusion of the unknown wordpiece.

Tab. 3 presents the UNK token percentage for each dataset using the MBERT vocabulary, averaged over each script and language type. This vocabulary is used in LAPT and represents the baseline level of vocabulary coverage. We also include the change in the UNK token percentage between the MBERT and VA vocabularies, which quantifies the coverage improvement. Both sets of values are juxtaposed against the average change in task-specific performance from LAPT to VA, representing the effect of augmenting the vocabulary on task-specific performance.

We observe that off-the-shelf MBERT already attains relatively high vocabulary coverage for Type 0 and 1 languages, as well as languages written in Latin and Cyrillic scripts. On the other hand, up to one-fifth of the tokens in Arabic languages and one-sixth of those in Type 2 languages yield an unknown wordpiece. For these languages, there is great room for increasing tokenizer coverage, and VA indeed addresses this more tangible need. This aligns with the task-specific performance improvements for each group and helps to explain our results in §2.3.

It is notable that VA does not always eliminate the issue of unknown wordpieces, even in languages for which MBERT attains high vocabulary coverage. This suggests that the remaining unknown wordpieces in these languages are more sparsely distributed (i.e., they represent low frequency sequences), while the unknown wordpieces in languages with lower vocabulary coverage represent sequences that occur more commonly. As a result, augmenting the vocabulary in such languages quickly improves coverage while associating these commonly occurring sequences with each other, which benefits the overall tokenization quality.

We further explore the association between the improvements in vocabulary coverage and task-specific performance in Fig. 1. Although we do not find that languages from the same types or scripts form clear clusters, we nonetheless observe a loose

Lang. Group (# of Langs.)	Avg. UNK Token % (MBERT) Unlabeled	UD	WikiAnn	Avg. UNK Token % (Δ) Unlabeled	UD	WikiAnn	Avg. Task Performance (Δ) POS	UD	NER
All (9)	5.9 % (–)	5.2 % (–)	6.2 % (–)	–5.3 % (–)	4.7 % (–)	–5.8 % (–)	+1.45 (–)	+2.08 (–)	+2.02 (–)
Type 0 (4)	1.0 % (↓)	0.3 % (↓)	1.2 % (↓)	–0.9 % (↑)	–0.3 % (↑)	–1.2 % (↑)	–0.13 (↓)	–0.04 (↓)	–0.05 (↓)
Type 1 (1)	0.3 % (↓)	0.0 % (↓)	0.4 % (↓)	–0.3 % (↑)	–0.00 % (↑)	–0.4 % (↑)	+0.05 (↓)	–0.44 (↓)	+0.69 (↓)
Type 2 (4)	12.3 % (↑)	13.5 % (↑)	14.8 % (↑)	–10.8 % (↓)	–12.1 % (↓)	–13.7 % (↓)	+4.03 (↑)	+5.74 (↑)	+5.23 (↑)
Latin (4)	1.2 % (↓)	0.6 % (↓)	2.4 % (↓)	–1.2 % (↑)	–0.6 % (↑)	–2.3 % (↑)	+0.09 (↓)	–0.15 (↓)	–0.27 (↓)
Cyrillic (3)	3.6 % (↓)	0.6 % (↓)	2.8 % (↓)	–3.6 % (↑)	–0.6 % (↑)	–2.7 % (↑)	–0.21 (↓)	+0.14 (↓)	+1.52 (↓)
Arabic (2)	19.0 % (↑)	19.2 % (↑)	16.9 % (↑)	–16.1 % (↓)	–17.0 % (↓)	–15.5 % (↓)	+5.84 (↑)	+8.49 (↑)	+6.22 (↑)

Table 3: Average UNK token percentage under the MBERT vocabulary (left); change in UNK token percentage from MBERT to VA vocabularies (center); and average task performance change from LAPT to VA (right). Averages are computed overall and within each script and language type, with comparisons to the overall average; all UNK token percentages are computed on the respective training sets for illustration. Note that Uyghur accounts for a large portion of the behavior of the Type 2/Arabic rows.

correlation between the two factors in question and see that VA delivers greater performance gains on Type 2 and Arabic-script languages compared to their Type 0/1 and Latin-script counterparts, respectively. To quantify the strength of this association, we also compute the language-level Spearman correlation between the change in UNK token percentage on the unlabeled dataset[7] from the MBERT to VA vocabulary and the task-specific performance improvements from LAPT to VA. The resulting ρ-values – 0.29 for NER, 0.56 for POS tagging, and 0.81 for UD parsing – suggest that this set of factors is meaningful for some tasks, though additional and more fine-grained analysis in future work should give a more complete explanation.

3 Mix-in Specialization: VA and Transliteration

We now expand on the observation made in §2.3 regarding the difficulties that MBERT encounters when faced with unseen low-resource languages in non-Latin scripts because of its inability to model the script. Having observed that VA is beneficial in such cases, we now investigate the interaction between this method and another specialization approach that targets this problem. Specifically, we consider the transliteration methods of Muller et al. (2021), in which unseen low-resource languages in non-Latin scripts are transliterated into the Latin script, often using transliteration schemes inspired by the Latin orthographies of languages related to the target language. They hypothesize that the increased similarity in the languages' writing systems, combined with MBERT's overall Latin-centricity, provides increased opportunity for crosslingual transfer.

We can view transliteration as a inverted form of vocabulary augmentation: instead of adapting the model to the needs of the data, the data is adjusted to meet the assumptions of the model. Furthermore, the transliteration step is performed prior to pretraining MBERT on additional unlabeled data in the target language, the same stage at which VA is performed. In both cases, the ultimate goal is identical: improving tokenization and more effectively using available data. We can thus view transliteration and VA as two instantiations of a more general *mix-in* paradigm for model specialization, whereby various transformations (mix-ins) are applied to the data and/or model prior to performing additional pretraining. These mix-ins target different components of the experimental pipeline, which naturally raises our second research question:

RQ2: How do the VA and transliteration mix-ins for MBERT compare and interact?

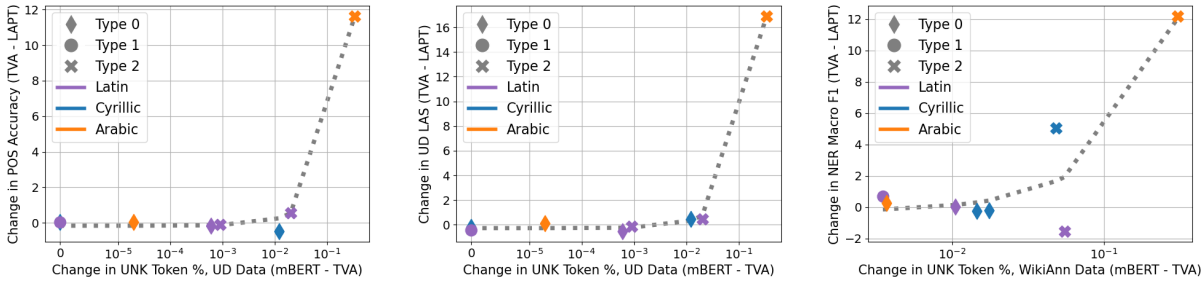
3.1 Method and Experiments

To test this research question, we apply transliteration and VA in succession and evaluate their compatibility. Given unlabeled data in the target language, we first transliterate it into Latin script, which decreases but does not fully eliminate the issue of unseen wordpieces. We then perform VA, generating the vocabulary for augmentation based on the transliterated data.

We evaluate on Meadow Mari and Uyghur, which are Type 2 languages where transliteration was successfully applied by Muller et al. (2021). To transliterate the data, we use the same methods as Muller et al. (2021): Meadow Mari uses the `transliterate`[8] package, while Uyghur uses

[7] We benchmark against the unlabeled dataset instead of task-specific ones for comparability.

[8] https://pypi.org/project/transliterate



(a) POS tagging. (b) UD parsing. (c) NER.

Figure 1: Relationship between the change in UNK token percentage on task data and the change in task performance, from (MBERT/LAPT to VA), with a 1-degree line of best fit. All vocabulary values are computed on the respective training sets.

a linguistically-motivated transliteration scheme[9] aimed at associating Uyghur with Turkish. We use the same training scheme, model architectures, and baselines as in §2.2, the only difference being the use of transliterated data. This includes directly pretraining on the unlabeled data (LAPT), which is comparable to the highest-performing transliteration models of Muller et al. (2021). Although our initial investigation of VA in §2 also included non-Type 2 languages of other scripts, we omit them from our investigation based on the finding of Muller et al. (2021) that transliterating higher-resource languages into Latin scripts is not beneficial.

3.2 Results

Tab. 4 gives the results of our transliteration mix-in experiments. For the MBERT-based models, both VA and transliteration provide strong improvements over their respective baselines. Specifically, the improvements from LAPT to VA and LAPT to LAPT with transliteration are most pronounced. This verifies the independent results of Chau et al. (2020) and Muller et al. (2021) and suggests that in the non-Latin low-resource setting, unadapted additional pretraining is insufficient, but that the mix-in stage between initial and additional pretraining is amenable to performance-improving modifications. Unsurprisingly, transliteration provides no consistent improvement to the monolingual baselines, since the noisy transliteration process removes information without improving crosslingual alignment.

However, VA and transliteration appear to interact negatively. Although VA with transliteration improves over plain VA for Uyghur POS tagging and dependency parsing, it still slightly underperforms LAPT with transliteration for the latter. For the two NER experiments, VA with transliteration lags both methods independently. One possible explanation is that transliteration into Latin script serves as implicit vocabulary augmentation, with embeddings that have already been updated during the initial pretraining stage; as a result, the two sources of augmentation conflict. Alternatively, since the transliteration process merges certain characters that are distinct in the original script, VA may augment the vocabulary with misleading character clusters. Either way, additional vocabulary augmentation is generally not as useful when combined with transliteration, answering **RQ2**.

Nonetheless, additional investigation into the optimal amount of vocabulary augmentation might yield a configuration that is consistently complementary to transliteration and is an interesting direction for future work. Furthermore, designing linguistically-informed transliteration schemes like those devised by Muller et al. (2021) for Uyghur requires large amounts of time and domain knowledge. VA's fully data-driven nature and relatively comparable performance suggest that it achieves an appealing balance between performance gain and implementation difficulty.

4 Related Work

Our work follows a long line of studies investigating the performance of multilingual language models like MBERT in various settings. The exact source of such models' crosslingual ability is contested: early studies attributed MBERT's success to vocabulary overlap between languages (Cao et al., 2020; Pires et al., 2019; Wu and Dredze, 2019),

[9] https://github.com/benjamin-mlr/mbert-unseen-languages

Rep.	MHR (NER)		UG (NER)		UG (POS)		UG (UD)	
FASTT	35.28 → 41.32	(+6.04)	–		89.45 → 89.03	(–0.42)	54.52 → 54.45	(–0.07)
BERT	54.17 → 48.45	(–5.72)	61.54 → 63.05	(+1.51)	90.87 → 90.76	(–0.09)	60.34 → 60.08	(–0.26)
MBERT	61.85 → 63.84	(+1.99)	50.76 → 56.80	(+6.04)	78.07 → 91.34	(+13.27)	47.70 → 65.85	(+18.15)
LAPT	59.17 → 63.68	(+4.51)	56.76 → 67.57	(+10.81)	79.88 → 92.59	(+12.71)	50.67 → **69.39**	(+18.72)
VA	**64.23** → 63.19	(–1.04)	**68.93** → 67.10	(–1.83)	91.49 → **92.64**	(+1.15)	67.55 → 68.58	(+1.03)

Table 4: Comparison of model performance before and after transliteration. **Bolded** results are the maximum for each language-task pair. – indicates that a model did not converge.

but subsequent studies find typological similarity and parameter sharing to be better explanations (Conneau et al., 2020b; K et al., 2020). Nonetheless, past work has consistently highlighted the limitations of multilingual models in the context of low-resource languages. Conneau et al. (2020a) highlight the tension between crosslingual transfer and per-language model capacity, which poses a challenge for low-resource languages that require both. Indeed, Wu and Dredze (2020) find that MBERT is unable to outperform baselines in the lowest-resource seen languages. Our experiments build off these insights, which motivate the development of methods for adapting MBERT to target low-resource languages.

Adapting Language Models Several prior studies have proposed methods for adapting pretrained models to a downstream task. The simplest of these is to perform additional pretraining on unlabeled data in the target language (Chau et al., 2020; Muller et al., 2021; Pfeiffer et al., 2020), which in turn builds off similar approaches for domain adaptation (Gururangan et al., 2020; Han and Eisenstein, 2019). Recent work uses one or more of these additional pretraining stages to specifically train modular adapter layers for specific tasks or languages, with the goal of maintaining a language-agnostic model while improving performance on individual languages (Pfeiffer et al., 2020, 2021a; Vidoni et al., 2020). However, as Muller et al. (2021) note, the typological diversity of the world's languages ultimately limits the viability of this approach.

On the other hand, many adaptation techniques have focused on improving representation of the target language by modifying the model's vocabulary or tokenization schemes (Chung et al., 2020; Clark et al., 2021; Wang et al., 2021). This is well-motivated: Artetxe et al. (2020) emphasize representation in the vocabulary as a key factor for effective crosslingual transfer, while Rust et al. (2021) find that MBERT's tokenization scheme for many languages is subpar. Pfeiffer et al. (2021b) further observe that for languages with unseen scripts, a large proportion of the language is mapped to the generic "unknown" wordpiece, and they propose a matrix factorization-based approach to improve script representation. Wang et al. (2020) extend MBERT's vocabulary with an entire new vocabulary in the target language to facilitate zero-shot transfer to low-resource languages from English. The present study most closely derives from Chau et al. (2020), who select 99 wordpieces with the greatest amount of coverage to augment MBERT's vocabulary while preserving the remainder; and Muller et al. (2021), who transliterate target language data into Latin script to improve vocabulary coverage. We deliver new insights on the effectiveness and applicability of these methods.

5 Conclusion

We explore the interactions between vocabulary augmentation and script transliteration for specializing multilingual contextual word representations in low-resource settings. We confirm vocabulary augmentation's effectiveness on multiple languages, scripts, and tasks; identify the mix-in stage as amenable to specialization; and observe a negative interaction between vocabulary augmentation and script transliteration. Our findings highlight several open questions in model specialization and low-resource natural language processing at large, motivating further study in this area.

Future directions for investigation are manifold. In particular, our results in this work unify the separate findings of past works, which use MBERT as a case study; a natural continuation would extend these methods to a broader set of multilingual models, such as mT5 (Xue et al., 2021) and XLM-R (Conneau et al., 2020a), in order to obtain a clearer understanding of the factors behind specialization methods' patterns of success. While we intentionally choose a set of small unlabeled datasets to evaluate on a setting applicable to the vast majority of the world's low-resource languages, we acknowl-

edge great variation in the amount of unlabeled data available in different languages. Continued study on the applicability of these methods to datasets of different sizes is an important future step. An interesting direction of work is to train multilingual models on data where script respresentation is more balanced, which might also allow for different output scripts for transliteration. Given that the mix-in stage is an effective opportunity to specialize models to target languages, constructing mix-ins at both the data and model level that are complementary by design has potential to be beneficial. Finally, future work might shed light on the interaction between different configurations of the adaptations studied here (e.g., the number of wordpiece types used in vocabulary augmentation).

Acknowledgments

We thank Jungo Kasai, Phoebe Mulcaire, and members of UW NLP for their helpful comments on preliminary versions of this paper. We also thank Benjamin Muller for insightful discussions and providing details about transliteration methods and baselines. Finally, we thank the anonymous reviewers for their helpful remarks.

References

Judit Ács. 2019. Exploring BERT's vocabulary.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proc. of ACL*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *Proc. of ICLR*.

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 ACL: EMNLP*.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proc. of EMNLP*.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. Canine: Pre-training an efficient tokenization-free encoder for language representation.

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 *Proc. of ACL*.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proc. of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proc. of NLP-OSS*.

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 *Proc. of ACL*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proc. of EMNLP-IJCNLP*.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pretraining with multiple cross-lingual tasks. In *Proc. of EMNLP-IJCNLP*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *Proc. of ICLR*.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proc. of EMNLP-IJCNLP*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proc. of NeurIPS*.

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

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 *Proc. of NAACL-HLT*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proc. of LREC*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proc. of ACL*.

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

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. AdapterFusion: Non-destructive task composition for transfer learning. In *Proc. of EACL*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proc. of EMNLP*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. Unks everywhere: Adapting multilingual language models to new scripts.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proc. of ACL*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proc. of ACL*.

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 *Proc. of ACL-IJCNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proc. of NAACL-HLT*.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of ACL: EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proc. of EMNLP*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proc. of EMNLP-IJCNLP*.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proc. of RepL4NLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

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

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga

Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.