Script, Language, and Labels: Overcoming Three Discrepancies for Low-Resource Language Specialization

Jaeseong Lee, Dohyeon Lee and Seung-won Hwang*

Computer Science and Engineering, Seoul National University
{tbvj5914,waylight3,seungwonh}@snu.ac.kr

Abstract

Although multilingual pretrained models (mPLMs) enabled support of various natural language processing in diverse languages, its limited coverage of 100+ languages lets 6500+ languages remain 'unseen'. One common approach for an unseen language is specializing the model for it as target, by performing additional masked language modeling (MLM) with the target language corpus. However, we argue that, due to the discrepancy from multilingual MLM pretraining, a naïve specialization as such can be suboptimal. Specifically, we pose three discrepancies to overcome. **Script** and **linguistic** discrepancy of the target language from the related seen languages, hinder a positive transfer, for which we propose to maximize representation similarity, unlike existing approaches maximizing overlaps. In addition, **label** space for MLM prediction can vary across languages, for which we propose to reinitialize top layers for a more effective adaptation. Experiments over four different language families and three tasks shows that our method improves the task performance of unseen languages with statistical significance, while previous approach fails to.

1. Introduction

Recently, multilingual pretrained language models (mPLMs), such as mBERT (Devlin et al. 2019) or XLM-R (Conneau et al. 2020a), became a de-facto standard to support natural language processing (NLP) over diverse languages. These models are pretrained by masked language modeling (MLM) with multilingual corpora of 100+ languages and shared parameters, mapping diverse languages to a shared feature space. However, the majority of 6500+ languages inevitably remain 'unseen' by existing mPLMs, calling for approaches to prepare models supporting them.

Meanwhile, training a new mPLM or monolingual PLM is inappropriate. While training a new mPLM to include a new language seems to be attractive, capacity conflict among multiple languages–known as the curse of multilinguality (Conneau et al. 2020a)–makes this solution impractical. Training a monolingual PLM for such low-resource languages is even less practical: Its task performance is reportedly inferior to mPLM (Wu and Dredze 2020).

Therefore, specialization[1] of mPLM to the target unseen language has been proposed as a promising alternative. One common technique is performing adaptive pretraining (Gururangan et al. 2020) using a corpus of the new language (Chau, Lin, and Smith 2020).

Although it was effective for some cases, Muller et al. (2021a) inspected some languages that struggle to improve, and attributed the reason to 'script discrepancy' (e.g. Latin vs Arabic script) with their related languages[2] in the mPLM. Due to the script difference, they do not share tokens, thus the target language could not fully benefit from related languages via knowledge transfer. A naïve solution is to transliterate (Muller et al. 2021a) to enforce token overlap (Figure 1a), where the word in unseen language is replaced by its romanization.

However, transliteration does not outperform a simple solution (Figure 1b), such as vocabulary augmentation (Chau and Smith 2021). Moreover, there may exist unexplored discrepancies, impeding better specialization.

In this paper, we propose to examine **three** discrepancies–Script, Linguistic, and Label–between mPLM and its specialization, and provide simple yet effective solutions for them. For 'script discrepancy', we first argue that measuring it with token overlap is the main problem of transliteration. Sharing script via naïve transliteration has a positive effect of token overlaps, but also a negative effect of information loss (Amrhein and Sennrich 2020), when mapping two homophones to the same romanized word. Next, since token overlaps, encouraging shared scripts, fail to predict gains, we propose to measure the script discrepancy without such requirement, by repurposing representation dissimilarity (Kudugunta et al. 2019; Conneau et al. 2020b), used in the context of cross-lingual transfer. We optimize this metric by aligning the word in the unseen language with its transliteration, in the representation space, which we call 'cross-script alignment' (Fig. 1 solid line). In this way, we can close the gap between two different scripts, while avoiding the loss from sharing.

Although we remedy script discrepancy, we observe that

*Hwang is the corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] Following Chau and Smith (2021), we use word specialization as a special case of an adaptation, in the sense of considering the target language exclusively.

[2] Roughly defined as seen languages in the same family as the target

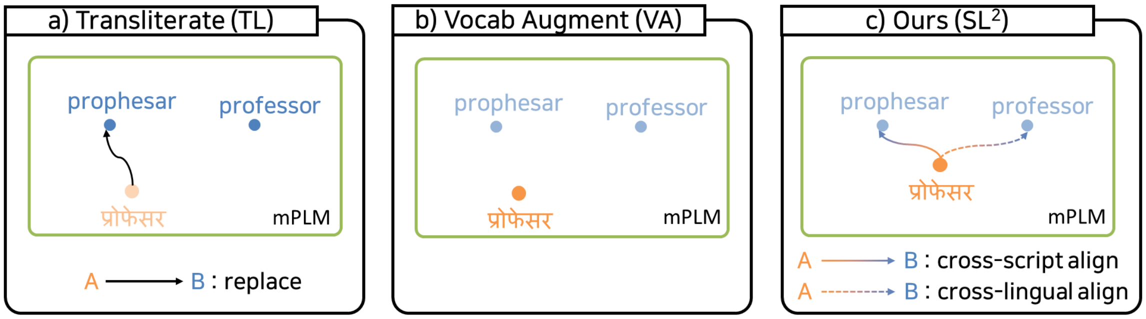

Figure 1: A hypothetical illustration of baselines and SL^2 dealing with script or linguistic discrepancies, between related language seen in multilingual pretraining (blue) and unseen language for specialization (orange).

discrepancy between the target language word and its semantic equivalent in related languages remains, which we call 'linguistic discrepancy'. For example, in Figure 1, even if the unseen word (shown in orange) benefits from cross-script alignment (solid line), representation dissimilarity remains between the word and its semantic equivalent, 'professor' (dotted line). For this discrepancy, we propose to add another type of alignment *before* specialization, which we show empirically better than alignments *after* specialization (Cao, Kitaev, and Klein 2020; Khemchandani et al. 2021) proposed for a different goal of cross-lingual transfer learning.

After overcoming the above two discrepancies in the input representation space, we move on to examine discrepancies in output labels, which we name as 'label discrepancy'. When mPLM is naïvely specialized to u, model parameters of mPLM are accustomed to the label prediction task of seen languages, such that mPLM would assign only a small probability to predict tokens of u (as some such tokens are seen as foreign words during training). Meanwhile, after specialization, such probability significantly increases, while the bias often holds it low, forcing the model to predict tokens from seen languages more frequently. We explore transfer learning techniques of preserving task-generic layers, while unlearning biases in task-specific layers.

In sum, we identify **S**cript, **L**inguistic, and **L**abel discrepancies (SL^2) in the view of both representations of input data and the output labels, and provide efficient remedies for them. Our contributions can be summarized as follows:

- We provide a different metric for previously proposed script discrepancy, and propose a novel concept of 'cross-script alignment' to mitigate it.
- We reveal two more discrepancies (linguistic and label) and provide simple yet effective solutions for them.
- Our method significantly outperforms baselines, in four diverse languages and three different tasks, while the previous approach to mitigate discrepancies fails to.
- Code and datasets we used are available.[3]

2. Proposed Method

In this section, we first review multilingual pretraining, and specialization of mPLM. Then, regarding the inputs, we pose script and linguistic discrepancies between them, then illustrate remedies in the view of representation similarity with related languages. Finally, regarding the output labels, we pose label discrepancy, and propose a simple yet effective solution to mitigate it.

2.1. Preliminaries

Multilingual Pretraining to build mPLM For each language l, and sentence S_l, the output of the transformer-based language model f_{θ,E_V} can be formulated as follows:

$$h_0 = E_V(\text{tok}_V(S_l)) \tag{1}$$

$$f^i_{\theta,E_V}(S_l) = h_i = L_i(h_{i-1}) \tag{2}$$

$$f_{\theta,E_V}(S_l) = f^N_{\theta,E_V}(S_l) \tag{3}$$

where V, E_V, and tok$_V$ denote the vocabulary, embedding layer of it, and tokenization process using V. L_i denotes the ith transformer layer, N denotes the number of layers, and θ denotes the union of parameters inside transformer layers.

For masked language modeling (MLM), some words $\{w^i_l\}_{i\in\mathbf{m}}$ are picked from S_l and replaced to mask tokens to generate S^m_l. Then, head layer W and a bias vector b are added to the top to design output as follows:

$$o = E^T_V(W f_{\theta,E_V}(S^m_l)) + b \tag{4}$$

which is consumed by cross-entropy loss with the labels of $\{w^i_l\}_{i\in\mathbf{m}}$. An mPLM is pretrained alternating these sentences S^m_l for multiple languages $l \in \mathcal{L}$.

Specialization of mPLM Even if we pretrain mPLM with $|\mathcal{L}| >= 100$, there exist languages not covered by such pretraining, and the performance of those is reportedly suboptimal (Muller et al. 2021a; Chau, Lin, and Smith 2020). A typical procedure to overcome is specializing mPLM by performing adaptive pretraining.

Let us denote the target unseen language as $u \notin \mathcal{L}$. Typically V needs to be specialized, since it lacks tokens of u, resulting in a high ratio of unknown tokens (Chau, Lin,

[3] https://github.com/thnkinbtfly/SL2

and Smith 2020; Chau and Smith 2021). Thus additional tokens V_u are selected from a newly trained wordpiece tokenizer for u, and used to augment the vocabulary of mPLM, $V' = V \cup V_u$. E is also augmented to E', with randomly initialized embedding for new tokens. Now equation 4 changes as follows:

$$o' = E'^{T}_{V'}(Wf_{\theta,E'_{V'}}(S^m_u)) + b \tag{5}$$

2.2. Discrepancies between Multilingual Pretraining and Specialization: Mitigation

We now describe the discrepancies between multilingual pretraining (Eq. 4) and specialization (Eq. 5), with how to alleviate them.

Script Discrepancy: Cross-Script Alignment First, we describe the *script discrepancy* as defined in the previous approach: to increase token overlap by transliteration. Unseen language u may not share the same script with related languages seen by mPLM. Previous works (Muller et al. 2021a) conjectured that such discrepancy makes specialization harder, and transliterated them to share the script with the related languages. However, increased token overlap by transliteration does not guarantee naturally higher performance (Chau and Smith 2021), and suboptimal performance of it is often attributed to an inherent characteristic of transliteration: It is not an injective mapping, leading to the collapse of scripts that were distinct in the original corpus (Chau and Smith 2021; Amrhein and Sennrich 2020).

Meanwhile, representation dissimilarity can be used to mitigate discrepancy without causing such a collapse. For simplicity, let us choose a representative related language $r \in \mathcal{L}$. When u uses a different script from r, the majority of $\text{tok}_{V'}(S_u)$ cannot be shared with $\text{tok}_{V'}(S_r)$, which results in dissimilar representations. With transliterator (Muller et al. 2021a), we convert the script of u into transliteration $\hat{r}$, which uses the same script as r. Now the $\text{tok}_{V'}(S_{\hat{r}})$ are more likely to be shared with $\text{tok}_{V'}(S_r)$, resulting in similar representations. In contrast, we apply 'cross-script alignment', to align the representations of $S_u = \{w^i_u\}$ to be similar to the representation of $S_{\hat{r}} = \{w^i_{\hat{r}}\}$. To align embeddings of two different scripts, we optimize the following equation, by re-interpreting an existing cross-lingual alignment method (Cao, Kitaev, and Klein 2020; Kulshreshtha, Redondo Garcia, and Chang 2020):[4]

$$\max_{\theta} \sum_{k \in K} \sum_{i} sim(f^k_{\theta,E'_{V'}}(w^i_u), f^k_{\theta_0,E'_{V'}}(w^i_{\hat{r}})) \tag{6}$$

where sim indicates the similarity metric, and K denotes the set of layer indices whose output is the optimization target. We let θ_0 fixed with the initial parameters of mPLM, since our goal is to shift the script embeddings of unseen language to be similar to the script embeddings of related languages in mPLM. Consequently, the representation of S_u would become similar to the representation of $S_{\hat{r}}$, which is more likely to be similar to the representation of S_r, providing a connection between u and r.

Linguistic Discrepancy: Cross-Lingual Alignment With our approach to measure discrepancy, we can observe another discrepancy. Although cross-script alignment bridges the gap between unseen language and related languages in mPLM, representation dissimilarity between $\hat{r}$ with the related languages r still remains, which we call *linguistic discrepancy*. For example, in Figure 1c, after cross-script alignment, there is a gap remaining to improve linguistic similarity by aligning the unseen word with its semantic counterpart.

Thus we propose to apply cross-lingual alignment before specialization, to further alleviate this discrepancy. Previous works of cross-lingual alignments (Cao, Kitaev, and Klein 2020; Khemchandani et al. 2021) can be interpreted as mitigating language discrepancy *after* specialization. However, such additional updates after language modeling confront a trade-off between mitigation and the performance of the language model. In contrast, we diminish this struggle, by avoiding further updates after language modeling of u.

Given parallel sentences S_u, S_r, we generate unsupervised word alignment $a = \{(i, j)\}$, which indicates w^i_u and w^j_r correspond each other semantically. Now equation 6 changes as follows:

$$\max_{\theta} \sum_{k \in K} \sum_{i,j \in a} sim(f^k_{\theta,E'_{V'}}(w^i_u), f^k_{\theta_0,E'_{V'}}(w^j_r)) \tag{7}$$

which would bridge the gap between u and r. Note that while this shares a similar goal with cross-script alignment, both two alignments complement to improve the performance, as we discuss in Section 3.

Label Discrepancy: Reinitialization of Head Finally, regarding output labels, *label discrepancy* may exist: Multilingual pretraining with $|\mathcal{L}|$ languages, when $u \notin \mathcal{L}$, rarely predicts u as output, as it is very rarely seen as a foreign token during training. However, after specialization to u, such probability should significantly increase, while the tendency to rarely predict u may remain, which would impede language modeling for u.

To mitigate, we propose to unlearn such a tendency by reinitializing W and b in equation 5, before performing specialization. As the latter layers are reported to be more task-specific (Zeiler and Fergus 2014; Muller et al. 2021b), reinitializing the latter layers would preserve general features in the former layers, while successfully unlearning the bias. We empirically found reinitialization of W and b, compared with L_is, is sufficient for this purpose, as we discuss in the next section.

3. Experiments

In this section, we describe experimental settings and conduct experiments to answer the following research questions:

- RQ1: Does SL^2 outperform baselines?
- RQ2: Is mitigation of each discrepancy essential?

[4] While Cao, Kitaev, and Klein (2020) introduces another term for regularization, we empirically found it not beneficial in our situation, which we discuss in supplementary materials.

Language (iso code)	Script	Family	Dominant Script	wiki size (MB)	word pairs (M)
Maltese (mt)	Latin	Afro-Asiatic	Arabic	8.89	0.350
Uyghur (ug)	Arabic	Turkic	Latin,Cyrillic	28.55	1.471
Erzya (myv)	Cyrillic	Uralic	Latin	4.59	0.002
Central Kurdish (ckb)	Arabic	Indo-European	Latin	43.36	0.091

Table 1: Languages considered in experiments

	ckb	mt		ug		myv	
	NER	POS	DEP	POS	DEP	POS	DEP
mBERT	77.92%	94.22%	76.46%	77.92%	47.70%	90.33%	67.06%
TL	87.80%	74.17%	44.84%	92.98%	69.64%	80.37%	56.29%
VA	88.52%	96.86%	82.84%	93.05%	69.70%	91.71%	72.19%
SL^2 (ours)	**89.49%**	**96.98%**	**83.35%**	**93.18%**	**70.54%**	91.50%	**73.21%**

Table 2: Comparison between proposed method and baselines. We report the average of F1 for NER, accuracy for POS, LAS for DEP. Statistically significant best results are bolded, while insignificant best results are underlined with the second best result (1-sided paired t-test, $p < 0.05$). Previous approach for discrepancies (TL) does not outperform a simpler approach with no mitigation (VA), while SL^2 successfully outperforms VA.

- RQ3: What if we mitigate linguistic discrepancy after specialization?
- RQ4: How should we select the number of layers to reinitialize in mitigation of label discrepancy?
- RQ5: Does the gain of SL^2 simply come from increase of data?
- RQ6: Is SL^2 effective on cross-lingual transfer?

3.1. Experimental Settings

We conduct experiments with mBERT pretrained using 104 largest Wikipedias (Devlin et al. 2019) as the representative mPLM, to be consistent with previous works (Muller et al. 2021a; Chau and Smith 2021).

Tasks and Labeled Datasets While tasks supporting low-resource languages are rare (Ahuja et al. 2022), tasks supporting unseen languages not covered by mPLMs are much scarcer. Following previous works covering unseen languages (Muller et al. 2021a; Chau and Smith 2021), we evaluate our method on three tasks, named-entity recognition (NER), part-of-speech tagging (POS), and dependency parsing (DEP). For NER, we utilize WikiAnn (Pan et al. 2017) with a balanced split (Rahimi, Li, and Cohn 2019). We use Universal Dependencies (Nivre et al. 2020) version 2.5 (Zeman et al. 2019) for POS and DEP. When there is only a test dataset available for our target language, we perform an 8-fold cross-validation with an isolated fold for the validation set, following Muller et al. (2021a).

Unlabeled Datasets for Specialization To be consistent with previous works (Chau and Smith 2021; Chau, Lin, and Smith 2020), we perform adaptive pretraining with Wikipedia articles extracted by WIKIEXTRACTOR,[5] using 80% of them only. Our split is provided with our code.

Languages To show the effectiveness of our mitigation of three discrepancies, we select unseen languages which use a different script from the majority of its related languages. For a more reliable evaluation, we only consider language covered by datasets with sufficient test examples,[6] which results in four languages to probe with: Central Kurdish (Indo-European), Uyghur (Turkic), Erzya (Uralic), and Maltese (Afro-Asiatic). We identify language family in Glottolog (Hammarström et al. 2021), and scripts used by each language in Wiktionary.[7] We describe these languages in Table 1.

Transliterators for Script-Discrepancy We choose the dominant script for each language family based on the number of Wikipedia articles, which is described in Table 1. We transliterate the original scripts to the dominant scripts. As a baseline transliterator, we use the Buckwalter Latin to Arabic transliteration module (Buckwalter 2002) using camel-tools (Obeid et al. 2020), for Maltese. For other languages, we follow Muller et al. (2021a) for transliteration. We transliterate Central Kurdish and Uyghur, with the transliterator from Muller et al. (2021a). For Erzya, we use the Russian transliteration module from TRANSLITERATE[8] to convert Cyrillic script to Latin.

Parallel Datasets for Language-Discrepancy To select r, we followed Muller et al. (2021a) for Uyghur and Central Kurdish, and we selected the language with the largest number of Wikipedia articles based on Wikimedia,[9] for other languages. We probed OPUS (Tiedemann 2012) to obtain the highest resource corpus of each language pair u-r. We use Tanzil for Uyghur-Turkish (Turkic), Mozilla-

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

[6] We selected 500 test examples as a criterion. The majority of our candidate language families had at least one language with such a dataset.

[7] https://www.wiktionary.org

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

[9] https://meta.wikimedia.org/wiki/List_of_Wikipedias

mitigate discrepancies			ckb	mt		ug		myv	
script	language	label	NER	POS	DEP	POS	DEP	POS	DEP
✓	✓	✓	89.49%	96.98%	83.35%	93.18%	70.54%	91.50%	73.21%
✗	✓	✓	88.79%	96.76%	83.58%	93.18%	69.92%	91.39%	72.82%
✓	✗	✓	89.31%	96.76%	83.20%	93.12%	70.62%	91.55%	72.59%
✓	✗	✗	88.81%	96.74%	82.91%	93.16%	70.07%	91.58%	72.29%

Table 3: Ablation study of removing each technique for mitigating discrepancies between multilingual pretraining and specialization.

l10n for Erzya-Finnish (Uralic), TICO-19 (Anastasopoulos et al. 2020) for Central Kurdish-Kurmanji Kurdish (Indo-European), and QED (Abdelali et al. 2014) for Maltese-Arabic (Afro-Asiatic).

Implementation Details For V_u, we follow Chau and Smith (2021) to train wordpiece with 5000 tokens and select tokens based on how much each token contributes to reducing unknown tokens. We introduce 3275 new tokens for Uyghur, 2901 for Central Kurdish, 2044 for Maltese, and 201 for Erzya, since adding more tokens did not improve the ratio of unknown tokens. We select K as the last four layers, sim as l2-norm, generate word alignments a utilizing FAST_ALIGN (Dyer, Chahuneau, and Smith 2013), and perform alignment for 1 epoch, following a previous successful cross-lingual alignment method (Kulshreshtha, Redondo Garcia, and Chang 2020). The number of generated word alignments is depicted in Table 1. We consume 8 sentences per batch.

Then, we perform specialization via adaptive pretraining for 20 epochs, with a batch size of 16, learning rate of 2e-5, warmup of 1000 steps, only using masked language modeling (MLM) loss, following Chau and Smith (2021). Adaptive pretraining is performed on TPUv2-8.

Finally, the fine-tuning settings are similar to those in Chau and Smith (2021): We compute the output of mPLM as a weighted sum of each activation h_is. We add a CRF layer for NER, linear projection for POS, biaffine attention (Dozat and Manning 2017) for DEP, atop the output. We use layer-wise gradual unfreezing, discriminative fine-tuning, inverse square-root learning rate decay with linear warmup. Fine-tuning is performed up to 200 epochs, with early stop based on validation performance. The implementation is based on AllenNLP (Gardner et al. 2018). We report F1 for NER, accuracy for POS, and LAS for DEP, following Chau and Smith (2021). All experiments are run 5 times and the average score is revealed, except for 8-fold experiments, where we run once per each fold and take the average.

3.2. Experimental Results

RQ1: Ours vs baselines We conduct experiments with the following three baselines:

- mBERT: Direct fine-tuning of mPLM, without any specialization.
- Vocabulary Augmentation (VA): Following Chau, Lin, and Smith (2020)[10], we augment vocabulary with tokens of unseen language, then perform adaptive pretraining, as we described in the previous section.
- Transliteration (TL): Following Muller et al. (2021a), we transliterate the corpus and task dataset, before applying adaptive pretraining. Transliteration is the only existing method dealing with discrepancies between multilingual pretraining and specialization, to the best of our knowledge.

Table 2 shows the result. First, we can observe that transliterator alone (TL) cannot outperform VA, for these four languages. This is also true for Uyghur (ug) and Central Kurdish (ckb), where transliteration was claimed to be effective (Muller et al. 2021a). This indicates that previous approaches to tackle the discrepancy between multilingual pretraining and specialization cannot be applied effectively.

In contrast, our method improves the performance with statistical significance ($p < 0.05$), even with these hard languages where TL suffers. The improvement is more significant among the tasks with a larger room for an increase. This clearly reveals that our method successfully mitigates the discrepancies, as intended, while previous approaches fail to do so.

RQ2: Effectiveness of Each Mitigation We now revisit gains from RQ1, by ablation study of each mitigation. We demonstrate that each mitigation of 1) script discrepancy, 2) linguistic discrepancy, and 3) label discrepancy is indispensable to achieve our performance.

First, by comparing the first and second rows in Table 3, we observe that 'cross-script alignment' to mitigate script discrepancy is essential. For example, when ablated, Central Kurdish loses 0.7% of F1, which is significant considering the gain from VA was 1%. Recalling that naïve transliteration (TL), the previous baseline to mitigate script discrepancy, was inferior to VA (Table 2), this indicates that cross-script alignment is an effective method to overcome script discrepancy. Moreover, these two rows justify that cross-lingual alignment alone cannot close the gap between unseen language and related language, demonstrating that cross-script alignment has a unique role. One exceptional case of performance degradation is the DEP performance of Maltese (mt). This is due to the significantly poor performance of transliteration as shown in Table 2, which we leave as future work to improve such a baseline.

[10] We do not apply another variant of using a larger learning rate to newly initialized embeddings, since it shows inferior performance in our case, which is described in the supplementary materials.

	ckb	mt		ug		myv	
mitigate	NER	POS	DEP	POS	DEP	POS	DEP
after	89.23%	96.78%	83.27%	**93.19%**	70.47%	**91.51%**	72.44%
before (ours)	**89.49%**	**96.98%**	**83.35%**	93.18%	**70.54%**	91.50%	**73.21%**

Table 4: Comparison between mitigating language discrepancy before and after specialization.

	ckb	mt		ug		myv	
reinit layers	NER	POS	DEP	POS	DEP	POS	DEP
L_9,L_{10},L_{11},L_{12},head	90.34%	97.04%	93.35%	92.19%	64.75%	84.50%	69.32%
L_{10},L_{11},L_{12},head	90.13%	96.97%	93.24%	92.30%	64.73%	84.74%	70.22%
L_{11},L_{12},head	90.25%	97.05%	93.45%	92.23%	65.03%	84.86%	70.19%
L_{12},head	90.25%	**97.14%**	93.42%	92.23%	65.11%	84.92%	**70.90%**
head	**90.53%**	97.11%	**93.54%**	**92.32%**	**65.21%**	**84.96%**	70.77%

Table 5: Comparison between reinitializing head only or more layers. We report the average of validation score. Best score is emphasized with bold.

		VA (data+)	SL^2
ckb	NER	88.32%	**89.49%**
mt	POS	96.30%	**96.98%**
	DEP	81.83%	**83.35%**
ug	POS	93.13%	**93.18%**
	DEP	69.52%	**70.54%**
myv	POS	91.18%	**91.50%**
	DEP	70.42%	**73.21%**

Table 6: Comparison when we allow the same data as we used for SL^2 to the best baseline. Best score is emphasized with bold.

		VA	SL^2
ckb	NER	75.61%	**75.70%**
mt	POS	77.31%	**78.28%**
	DEP	55.21%	**57.42%**
ug	POS	69.33%	**70.78%**
	DEP	31.41%	**33.25%**
myv	POS	71.56%	**73.60%**
	DEP	41.26%	**43.52%**

Table 7: Cross-lingual transfer performance comparison between SL^2 and the best baseline. Best score is emphasized with bold.

Second, when we analyze the first and third rows in Table 3, we can observe that cross-lingual alignment is playing another role. For example, we observe that POS and DEP performance in Maltese is decreased when cross-lingual alignment is ablated.

Finally, the third and last rows in Table 3 show that tackling label discrepancy with reinitialization of the head layer is effective. For example, the DEP performance of Uyghur or NER performance of Central Kurdish decreases notably, when we remove this mitigation.

In sum, we conclude that all three techniques are contributing to the performance of our proposed method.

RQ3: Language Discrepancy Before vs After Specialization While we align, *before* specialization, previous works of cross-lingual alignment (Cao, Kitaev, and Klein 2020; Khemchandani et al. 2021) apply it *after* specialization. To support our choice, we compare with SL^2 aligning *after* specialization. We adopt the equation from Cao, Kitaev, and Klein (2020) instead of equation 7, since it was empirically better, while keeping all other settings.

Table 4 shows SL^2 aligning *before* specialization shows superior performance. As we explained in Section 2, we attribute the inferiority of mitigating 'after' specialization to changes of parameters for mitigation updates (negatively affecting language model). In contrast, SL^2 avoids such negative effect, by optimizing for language model after the mitigation.

RQ4: Why Reinitialize the Head Only? While we proposed to reinitialize the head only, or reinitializing W and b in equation 5, we probe whether an alternative strategy of reinitializing more layers would be beneficial.

Table 5 shows that reinitializing more layers generally cannot improve performance. This indicates that the former layers successfully capture more general features for masked language modeling, while the head layer is more dedicated to predicting tokens of each language. We thus reside with our choice of the reinitializing head only.

RQ5: Does the Gain Come from Increase of Data? One may question whether the gain of SL^2 simply comes from the increase of the data. To find out, we allow the same data to VA, the best baseline from Table 2, by continually pretraining on the concatenation of original data, transliterations, and the parallel data. We use the same number of update steps as before. Table 6 shows that, even though we allow the same data, it is outperformed by SL^2. This indicates that the gain of SL^2 is not easily achievable by the simple increase of data.

RQ6: SL^2 for Cross-lingual Transfer While our main goal is in-language task performance, the improved target language performance may also benefit cross-lingual transfer performance. To evaluate, we train the model with English data only, and select the best model based on the development set of the target language, following Keung et al. (2020). Table 7 shows that SL^2 is effective on cross-lingual transfer also.

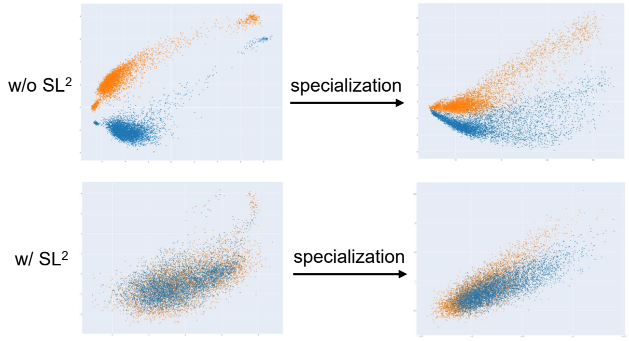

Figure 2: PCA visualization of sentence representations of Uyghur and its related language Turkish. Without our mitigations, Uyghur and Turkish seem to be dissimilar to mBERT, and the Hausdorff distance between the two languages was 0.54. With our method, mBERT successfully recognizes Uyghur as similar to Turkish throughout the specialization, and Hausdorff distance drops to 0.13.

3.3. In-Depth Analysis of Effectiveness of Our Mitigations: A Case Study with Uyghur

In this section, we present a deeper analysis of our proposed method. To show the benefit of SL^2 clearer, we take Uyghur as an unseen representative language, with a high-quality transliterator and large parallel resources available.

Representation Similarity We conjecture that our proposed techniques would improve the representation similarity between the two languages. To verify the conjecture, we conducted a PCA visualization of sentence representations with randomly sampled 5,000 Uyghur-Turkish parallel sentences. Moreover, we calculated the Hausdorff distance of sentence representations between two languages, following Xia et al. (2021). We consider [CLS] tokens as sentence representations, following previous works to compare sentence representations (Xia et al. 2021; Qin et al. 2020).

SL^2 shifts Uyghur to be similar to its highly-related language, Turkish (left of Figure 2). Moreover, this is true even after specialization (right of Figure 2). Hausdorff distance reverifies this argument. Without our mitigation, Hausdorff distance between [CLS] tokens of two languages remains at 0.54 after specialization. SL^2 drops this distance to 0.13. Note that we did not align special tokens such as [CLS]; We aligned corresponding words only, as described in the proposed method.

These imply SL^2 helps mPLM perceive Uyghur as similar to Turkish throughout the specialization stage, enhancing transfer from related languages, which is supported by improved performance in the experimental results.

Label Discrepancy Though the ablation study in Table 3 shows the performance gain from reinitialization, it is still unclear whether such gain comes from better language modeling for unseen language u. We thus further explore to confirm if (a) perplexity for u decreases, (b) token prediction from u increases, and (c) tokens unseen from u are less likely to be generated.

First, we compared the perplexity of specialized mPLM with and without our reinitialization, from the valid corpus of Chau and Smith (2021). Applying the reinitialization reduces the perplexity from 12.59 to 10.87, which shows that the reinitialized mPLM assigns higher probabilities to Uyghur tokens.

Second, we investigated the tendency to predict tokens that are more frequent in Uyghur. We collected tokens predicted differently by two models, from the same valid corpus. Then we calculated the average frequency of the tokens in the whole Uyghur corpus, containing train and test corpus from the same work. Reinitialization enhances the average frequency by 24.7%, indicating it helps the model output frequent tokens in Uyghur.

Finally, we analyzed the predictions of tokens that never occur in the whole Uyghur corpus. With reinitialization, the number of such predictions plummets to be 25 times lower.

These indicate that the original mPLM has a bias to predict tokens seen in the multilingual pretraining stage, and our solution to reinitialize successfully alleviates it.

4. Related Work

This section overviews the motivation and distinction from existing methods.

4.1. Why: Multilingual Pretraining Is Inappropriate to Cover All Languages

mPLMs are pretrained by MLM across the corpus of 100+ languages (Devlin et al. 2019; Conneau and Lample 2019; Conneau et al. 2020a). This maps these languages to a shared feature space, and the transfer of the features between related languages is prominent (Pires, Schlinger, and Garrette 2019; Wu and Dredze 2019).

However, mPLMs could not cover all of the 6500+ languages: Task performance of such 'unseen' languages is reportedly suboptimal (Muller et al. 2021a; Pfeiffer et al. 2020; Chau and Smith 2021). Moreover, building another mPLM supporting more languages is inappropriate. Curse of multilinguality (Conneau et al. 2020a) claimed that as more languages are trained with a fixed capacity, the performance of each language degrades, requiring even larger models. Wu and Dredze (2020) also argued that the performance of low-resource languages lags behind in mPLM. These call for an alternative direction to support unseen languages.

4.2. How: Specialization of mPLM to Unseen Language

Therefore, specializing mPLM has been proposed as a promising alternative (Chau, Lin, and Smith 2020). They continually pretrain mPLM with the corpus of unseen language in an unsupervised manner, often augmenting its vocabulary beforehand (Wang et al. 2020; Chau and Smith 2021; Chau, Lin, and Smith 2020). We take this approach as our baseline. Adapters (Houlsby et al. 2019; Pfeiffer et al. 2020), adding a few parameters for specialization and training them only, can be used additionally. Following Muller et al. (2021a), we do not consider adapters as our baseline methods, since they do not show significant performance improvement over simply using mPLM.

However they do not deal with discrepancies from multilingual pretraining, and we argue substantial improvement can be achieved when we consider them.

4.3. Distinction: Discrepancy between Multilingual Pretraining and Specialization

Overcoming discrepancies between pretrain-finetune discrepancy is important: Zhang et al. (2021) reinitialized the later layers in BERT, for example.

Similarly, overcoming script discrepancy between multilingual pretraining and specialization has attracted research interest. Muller et al. (2021a) inspected some languages struggle to improve performance by specialization, and attributed the reason to the script difference between the target language and related languages. They suggested converting the script to be the same as related languages by transliteration. RelateLM (Khemchandani et al. 2021) further aligns words between transliterated language and high-resource related language, after specialization finishes.

However, we argue that the previous solution for script discrepancy is suboptimal, and propose representation dissimilarity as a new metric to measure script discrepancy. We mitigate it with our novel technique 'cross-script alignment'. Moreover, we reveal that linguistic discrepancy remains, and further enhance similarity by mitigating it. Finally, we address label discrepancy as well, and provide a simple and effective solution of reinitializing the task-specific layer.

5. Conclusion

We studied the problem of low-resource language specialization, by identifying script, language, and label discrepancies as three main obstacles. We showed the limitation of existing solutions for minimizing discrepancies, such as transliteration aiming to deal with script overlaps, and proposed to maximize representation similarity in the script and language spaces, while preserving scripts. Our proposed solution, by maximizing new metrics, for script and language alignments, maximizes the transferability from mPLM, while unlearning its bias towards output labels seen in the training. We empirically validated the effectiveness of our approach, and showed each discrepancy counts toward such gains.

Acknowledgements

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2020-0-01789) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). This work was also supported by IITP grants funded by the MSIT [No.2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data], [NO.2021-0-0268, Artificial Intelligence Innovation Hub (Artificial Intelligence Institute, Seoul National University)], and [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]. We would like to thank Google's TPU Research Cloud (TRC) program for providing Cloud TPUs.

References

Abdelali, A.; Guzman, F.; Sajjad, H.; and Vogel, S. 2014. The AMARA Corpus: Building Parallel Language Resources for the Educational Domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Ahuja, K.; Dandapat, S.; Sitaram, S.; and Choudhury, M. 2022. Beyond Static Models and Test Sets: Benchmarking the Potential of Pre-Trained Models across Tasks and Languages. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*.

Amrhein, C.; and Sennrich, R. 2020. On Romanization for Model Transfer Between Scripts in Neural Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Anastasopoulos, A.; Cattelan, A.; Dou, Z.-Y.; Federico, M.; Federmann, C.; Genzel, D.; Guzmán, F.; Hu, J.; Hughes, M.; Koehn, P.; Lazar, R.; Lewis, W.; Neubig, G.; Niu, M.; Öktem, A.; Paquin, E.; Tang, G.; and Tur, S. 2020. TICO-19: The Translation Initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.

Buckwalter, T. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium. *University of Pennsylvania, LDC Catalog No.: LDC2002L49*.

Cao, S.; Kitaev, N.; and Klein, D. 2020. Multilingual Alignment of Contextual Word Representations. In *International Conference on Learning Representations*.

Chau, E. C.; Lin, L. H.; and Smith, N. A. 2020. Parsing with Multilingual BERT, a Small Corpus, and a Small Treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Chau, E. C.; and Smith, N. A. 2021. Specializing Multilingual Language Models: An Empirical Study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020a. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Conneau, A.; and Lample, G. 2019. Cross-Lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*.

Conneau, A.; Wu, S.; Li, H.; Zettlemoyer, L.; and Stoyanov, V. 2020b. Emerging Cross-lingual Structure in Pretrained

Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Dozat, T.; and Manning, C. D. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Dyer, C.; Chahuneau, V.; and Smith, N. A. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 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*.

Hammarström, H.; Forkel, R.; Haspelmath, M.; and Bank, S. 2021. Glottolog/Glottolog: Glottolog Database 4.5.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; Laroussilhe, Q. D.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*.

Keung, P.; Lu, Y.; Salazar, J.; and Bhardwaj, V. 2020. Don't Use English Dev: On the Zero-Shot Cross-Lingual Evaluation of Contextual Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Khemchandani, Y.; Mehtani, S.; Patil, V.; Awasthi, A.; Talukdar, P.; and Sarawagi, S. 2021. Exploiting Language Relatedness for Low Web-Resource Language Model Adaptation: An Indic Languages Study. 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)*.

Kudugunta, S.; Bapna, A.; Caswell, I.; and Firat, O. 2019. Investigating Multilingual NMT Representations at Scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Kulshreshtha, S.; Redondo Garcia, J. L.; and Chang, C.-Y. 2020. Cross-Lingual Alignment Methods for Multilingual BERT: A Comparative Study. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Muller, B.; Anastasopoulos, A.; Sagot, B.; and Seddah, D. 2021a. 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*.

Muller, B.; Elazar, Y.; Sagot, B.; and Seddah, D. 2021b. First Align, Then Predict: Understanding the Cross-Lingual Ability of Multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Nivre, J.; de Marneffe, M.-C.; Ginter, F.; Hajič, J.; Manning, C. D.; Pyysalo, S.; Schuster, S.; Tyers, F.; and Zeman, D. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*.

Obeid, O.; Zalmout, N.; Khalifa, S.; Taji, D.; Oudah, M.; Alhafni, B.; Inoue, G.; Eryani, F.; Erdmann, A.; and Habash, N. 2020. CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*.

Pan, X.; Zhang, B.; May, J.; Nothman, J.; Knight, K.; and Ji, H. 2017. Cross-Lingual Name Tagging and Linking for 282 Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Pfeiffer, J.; Vulić, I.; Gurevych, I.; and Ruder, S. 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)*.

Pires, T.; Schlinger, E.; and Garrette, D. 2019. How Multilingual Is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Qin, L.; Ni, M.; Zhang, Y.; and Che, W. 2020. CoSDA-ML: Multi-Lingual Code-Switching Data Augmentation for Zero-Shot Cross-Lingual NLP. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*.

Rahimi, A.; Li, Y.; and Cohn, T. 2019. Massively Multilingual Transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Tiedemann, J. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*.

Wang, Z.; K, K.; Mayhew, S.; and Roth, D. 2020. Extending Multilingual BERT to Low-Resource Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Wu, S.; and Dredze, M. 2019. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Wu, S.; and Dredze, M. 2020. Are All Languages Created Equal in Multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*.

Xia, M.; Zheng, G.; Mukherjee, S.; Shokouhi, M.; Neubig, G.; and Awadallah, A. H. 2021. MetaXL: Meta Representation Transformation for Low-resource Cross-lingual Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*.

Zeman, D.; Nivre, J.; Abrams, M.; Aepli, N.; Agić, Ž.; Ahrenberg, L.; Aleksandravičiūtė, G.; Antonsen, L.; Aplonova, K.; Aranzabe, M. J.; Arutie, G.; Asahara, M.; Ateyah, L.; Attia, M.; Atutxa, A.; Augustinus, L.; Badmaeva, E.; Ballesteros, M.; Banerjee, E.; Bank, S.; Barbu Mititelu, V.; Basmov, V.; Batchelor, C.; Bauer, J.; Bellato, S.; Bengoetxea, K.; Berzak, Y.; Bhat, I. A.; Bhat, R. A.; Biagetti, E.; Bick, E.; Bielinskienė, A.; Blokland, R.; Bobicev, V.; Boizou, L.; Borges Völker, E.; Börstell, C.; Bosco, C.; Bouma, G.; Bowman, S.; Boyd, A.; Brokaitė, K.; Burchardt, A.; Candito, M.; Caron, B.; Caron, G.; Cavalcanti, T.; Cebiroğlu Eryiğit, G.; Cecchini, F. M.; Celano, G. G. A.; Čéplö, S.; Cetin, S.; Chalub, F.; Choi, J.; Cho, Y.; Chun, J.; Cignarella, A. T.; Cinková, S.; Collomb, A.; Çöltekin, Ç.; Connor, M.; Courtin, M.; Davidson, E.; de Marneffe, M.-C.; de Paiva, V.; de Souza, E.; Diaz de Ilarraza, A.; Dickerson, C.; Dione, B.; Dirix, P.; Dobrovoljc, K.; Dozat, T.; Droganova, K.; Dwivedi, P.; Eckhoff, H.; Eli, M.; Elkahky, A.; Ephrem, B.; Erina, O.; Erjavec, T.; Etienne, A.; Evelyn, W.; Farkas, R.; Fernandez Alcalde, H.; Foster, J.; Freitas, C.; Fujita, K.; Gajdošová, K.; Galbraith, D.; Garcia, M.; Gärdenfors, M.; Garza, S.; Gerdes, K.; Ginter, F.; Goenaga, I.; Gojenola, K.; Gökırmak, M.; Goldberg, Y.; Gómez Guinovart, X.; González Saavedra, B.; Griciūtė, B.; Grioni, M.; Grūzītis, N.; Guillaume, B.; Guillot-Barbance, C.; Habash, N.; Hajič, J.; Hajič jr., J.; Hämäläinen, M.; Hà Mỹ, L.; Han, N.-R.; Harris, K.; Haug, D.; Heinecke, J.; Hennig, F.; Hladká, B.; Hlaváčová, J.; Hociung, F.; Hohle, P.; Hwang, J.; Ikeda, T.; Ion, R.; Irimia, E.; Ishola, Ọ.; Jelínek, T.; Johannsen, A.; Jørgensen, F.; Juutinen, M.; Kaşıkara, H.; Kaasen, A.; Kabaeva, N.; Kahane, S.; Kanayama, H.; Kanerva, J.; Katz, B.; Kayadelen, T.; Kenney, J.; Kettnerová, V.; Kirchner, J.; Klementieva, E.; Köhn, A.; Kopacewicz, K.; Kotsyba, N.; Kovalevskaitė, J.; Krek, S.; Kwak, S.; Laippala, V.; Lambertino, L.; Lam, L.; Lando, T.; Larasati, S. D.; Lavrentiev, A.; Lee, J.; Lê H`ông, P.; Lenci, A.; Lertpradit, S.; Leung, H.; Li, C. Y.; Li, J.; Li, K.; Lim, K.; Liovina, M.; Li, Y.; Ljubešić, N.; Loginova, O.; Lyashevskaya, O.; Lynn, T.; Macketanz, V.; Makazhanov, A.; Mandl, M.; Manning, C.; Manurung, R.; Mărănduc, C.; Mareček, D.; Marheinecke, K.; Martínez Alonso, H.; Martins, A.; Mašek, J.; Matsumoto, Y.; McDonald, R.; McGuinness, S.; Mendonça, G.; Miekka, N.; Misirpashayeva, M.; Missilä, A.; Mititelu, C.; Mitrofan, M.; Miyao, Y.; Montemagni, S.; More, A.; Moreno Romero, L.; Mori, K. S.; Morioka, T.; Mori, S.; Moro, S.; Mortensen, B.; Moskalevskyi, B.; Muischnek, K.; Munro, R.; Murawaki, Y.; Müürisep, K.; Nainwani, P.; Navarro Horñiacek, J. I.; Nedoluzhko, A.; Nešpore-Bērzkalne, G.; Nguy˜ên Thị, L.; Nguy˜ên Thị Minh, H.; Nikaido, Y.; Nikolaev, V.; Nitisaroj, R.; Nurmi, H.; Ojala, S.; Ojha, A. K.; Olúòkun, A.; Omura, M.; Osenova, P.; Östling, R.; Øvrelid, L.; Partanen, N.; Pascual, E.; Passarotti, M.; Patejuk, A.; Paulino-Passos, G.; Peljak-Łapińska, A.; Peng, S.; Perez, C.-A.; Perrier, G.; Petrova, D.; Petrov, S.; Phelan, J.; Piitulainen, J.; Pirinen, T. A.; Pitler, E.; Plank, B.; Poibeau, T.; Ponomareva, L.; Popel, M.; Pretkalniņa, L.; Prévost, S.; Prokopidis, P.; Przepiórkowski, A.; Puolakainen, T.; Pyysalo, S.; Qi, P.; Rääbis, A.; Rademaker, A.; Ramasamy, L.; Rama, T.; Ramisch, C.; Ravishankar, V.; Real, L.; Reddy, S.; Rehm, G.; Riabov, I.; Rießler, M.; Rimkutė, E.; Rinaldi, L.; Rituma, L.; Rocha, L.; Romanenko, M.; Rosa, R.; Rovati, D.; Roșca, V.; Rudina, O.; Rueter, J.; Sadde, S.; Sagot, B.; Saleh, S.; Salomoni, A.; Samardžić, T.; Samson, S.; Sanguinetti, M.; Särg, D.; Saulīte, B.; Sawanakunanon, Y.; Schneider, N.; Schuster, S.; Seddah, D.; Seeker, W.; Seraji, M.; Shen, M.; Shimada, A.; Shirasu, H.; Shohibussirri, M.; Sichinava, D.; Silveira, A.; Silveira, N.; Simi, M.; Simionescu, R.; Simkó, K.; Šimková, M.; Simov, K.; Smith, A.; Soares-Bastos, I.; Spadine, C.; Stella, A.; Straka, M.; Strnadová, J.; Suhr, A.; Sulubacak, U.; Suzuki, S.; Szántó, Z.; Taji, D.; Takahashi, Y.; Tamburini, F.; Tanaka, T.; Tellier, I.; Thomas, G.; Torga, L.; Trosterud, T.; Trukhina, A.; Tsarfaty, R.; Tyers, F.; Uematsu, S.; Urešová, Z.; Uria, L.; Uszkoreit, H.; Utka, A.; Vajjala, S.; van Niekerk, D.; van Noord, G.; Varga, V.; Villemonte de la Clergerie, E.; Vincze, V.; Wallin, L.; Walsh, A.; Wang, J. X.; Washington, J. N.; Wendt, M.; Williams, S.; Wirén, M.; Wittern, C.; Woldemariam, T.; Wong, T.-s.; Wróblewska, A.; Yako, M.; Yamazaki, N.; Yan, C.; Yasuoka, K.; Yavrumyan, M. M.; Yu, Z.; Žabokrtský, Z.; Zeldes, A.; Zhang, M.; and Zhu, H. 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.

Zhang, T.; Wu, F.; Katiyar, A.; Weinberger, K. Q.; and Artzi, Y. 2021. Revisiting Few-Sample BERT Fine-Tuning. In *International Conference on Learning Representations*.