How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models

Phillip Rust[*1], **Jonas Pfeiffer**[*1],
Ivan Vulić[2], **Sebastian Ruder**[3], **Iryna Gurevych**[1]
[1]Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
[2]Language Technology Lab, University of Cambridge
[3]DeepMind
www.ukp.tu-darmstadt.de

Abstract

In this work we provide a *systematic empirical comparison* of pretrained multilingual language models versus their monolingual counterparts with regard to their monolingual task performance. We study a set of nine typologically diverse languages with readily available pretrained monolingual models on a set of five diverse monolingual downstream tasks. We first establish if a gap between the multilingual and the corresponding monolingual representation of that language exists, and subsequently investigate the reason for a performance difference. To disentangle the impacting variables, we train new monolingual models on the same data, but with different tokenizers, both the monolingual and the multilingual version. We find that while the pretraining data size is an important factor, the designated tokenizer of the monolingual model plays an equally important role in the downstream performance. Our results show that languages which are adequately represented in the multilingual model's vocabulary exhibit negligible performance decreases over their monolingual counterparts. We further find that replacing the original multilingual tokenizer with the specialized monolingual tokenizer improves the downstream performance of the multilingual model for almost every task and language.

1 Introduction

Following large Transformer-based language models (LMs) (Vaswani et al., 2017) pretrained for the English language (e.g., BERT, RoBERTa, T5) (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) on large corpora, similar monolingual language models have been introduced for other languages (Virtanen et al., 2019; Antoun et al., 2020; Martin et al., 2020, *inter alia*), offering previously unmatched performance on virtually all NLP tasks. Concurrently, massively multilingual pretrained models with the same architectures and training procedures, but covering more than 100 languages in a single model, have been proposed (e.g., multilingual BERT (mBERT), XLM-R, multilingual T5) (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2020).

The "industry" of pretraining and releasing new monolingual BERT models continues its operations despite the fact that the corresponding languages are already covered by multilingual models such as mBERT and XLM-R. The common argument justifying the need for monolingual variants is the assumption that multilingual models, due to suffering from the so-called curse of multilinguality (Conneau et al., 2020) (i.e., the lack of capacity to represent all languages in an equitable way), underperform monolingual models when applied to monolingual tasks (Virtanen et al., 2019; Antoun et al., 2020; Rönnqvist et al., 2019, *inter alia*). However, little to no compelling empirical evidence with rigorous experiments and fair comparisons have been presented so far to support or invalidate this strong claim. In this regard, much of the work proposing and releasing new monolingual models is grounded on anecdotal evidence, pointing to the positive results reported for other monolingual BERT models (de Vries et al., 2019; Virtanen et al., 2019; Antoun et al., 2020).

Monolingual BERT models are typically evaluated on downstream NLP tasks in their particular languages to demonstrate their effectiveness in comparison to previous monolingual models or mBERT (Virtanen et al., 2019; Antoun et al., 2020; Martin et al., 2020, *inter alia*). While these results do show that *certain* monolingual models *can* outperform mBERT in *certain* tasks, we hypothesize that this may substantially vary across different languages and language properties, tasks, pretrained models and their pretraining data, domain, and

[*]Both authors contributed equally to this work.

size. We further argue that conclusive evidence, either supporting or refuting the key hypothesis that monolingual models currently outperform multilingual models, necessitates an independent and controlled empirical comparison on a diverse set of languages and tasks.

While recent work has argued that mBERT is under-trained (Rönnqvist et al., 2019; Wu and Dredze, 2020), providing evidence of improved performance when training a monolingual model on more data, it is unclear if this is the only important factor that improves the performance of monolingual models. For instance, another contributing factor might be the limited vocabulary size of a multilingual model compared to the sum of tokens of all corresponding monolingual models. This provokes analyses on whether dedicated (i.e., language-specific) tokenizers of monolingual models also play a critical role.

Contributions. In summary, our contributions are as follows. **1)** We systematically compare monolingual versus multilingual pretrained language models for 9 typologically diverse languages on 5 structurally different tasks. **2)** We train new monolingual models on equally sized datasets but relying on different tokenizers (i.e., shared multilingual tokenizers versus dedicated language-specific ones) to disentangle the impact of pretraining data size versus tokenization on the downstream performance. **3)** We isolate factors that contribute to performance difference (e.g., tokenizers' "fertility", the number of unseen (sub)words, data size) and provide an in-depth analysis of the impact of these important factors on task performance. Finally, we hope that our findings offer informed guidance for training new multilingual models in the future.

2 Background and Related Work

Multilingual LMs. The wide usage of pretrained multilingual Transformer-based LMs has been instigated by the release of multilingual BERT (Devlin et al., 2019) which followed on the success of the monolingual English BERT model. mBERT adopted the same pretraining regime as monolingual BERT on concatenated Wikipedia data for 104 languages with the largest Wikipedias. Exponential smoothing was used when creating the subword vocabulary based on WordPieces (Wu et al., 2016) and a pretraining corpus. By oversampling underrepresented languages and undersampling overrepresented languages, the goal is to counteract the imbalance of pretraining data sizes. The final shared mBERT vocabulary comprises a total of 119,547 subword tokens.

Other multilingual model variants followed mBERT, such as XLM-R (Conneau et al., 2020) based on the monolingual RoBERTa model (which is also a variant of the original BERT model) (Liu et al., 2019). Concurrently, many studies started analyzing mBERT's and XLM-R's capabilities and limitations, finding that the multilingual models work surprisingly well for cross-lingual tasks, despite the fact that they do not rely on any direct cross-lingual supervision (e.g., parallel or comparable data, translation dictionaries) (Pires et al., 2019; Wu and Dredze, 2019; K et al., 2020).

However, recent work has also pointed to some fundamental limitations of the multilingual models. Conneau et al. (2020) observe that, for a fixed model capacity, adding new languages increases cross-lingual performance up to a certain point. After that point is reached, adding new languages deteriorates performance. This phenomenon, termed the *curse of multilinguality*, can be attenuated by increasing the capacity of the model (Artetxe et al., 2020; Pfeiffer et al., 2020c; Chau et al., 2020) or through additional training for particular language pairs (Pfeiffer et al., 2020c; Ponti et al., 2020). Another observation concerns substantially reduced cross-lingual and monolingual abilities of the models for resource-poor languages with smaller pretraining data (Wu and Dredze, 2020; Lauscher et al., 2020b). Those languages are effectively still underrepresented in the subword vocabulary and the model's shared representation space despite oversampling. In general, these findings indicate that it is (currently) not possible to represent (all) languages of the world in a single model.

Monolingual versus Multilingual LMs. New monolingual language-specific models also emerged for many languages, following BERT's architecture and pretraining procedure. For instance, there are monolingual BERT variants for Arabic (Antoun et al., 2020), French (Martin et al., 2020), Finnish (Virtanen et al., 2019), Dutch (de Vries et al., 2019), Italian (Polignano et al., 2019), to name only a few. Pyysalo et al. (2020) released 44 monolingual WikiBERT models trained on Wikipedia. However, only a few studies have thus far, either explicitly or implicitly, attempted to understand how monolingual and multilingual BERTs compare across different

languages. Here, we briefly summarize previous attempts to understand these differences.

Nozza et al. (2020) extracted task results from the respective papers on monolingual BERTs, and listed them on a dedicated webpage[1] to facilitate an overview of monolingual models, and their comparison to mBERT. However, they simply copy the scores reported in the papers which were obtained under diverse experimental setups and training conditions: they have not verified the scores nor have performed a controlled impartial comparison.

Vulić et al. (2020) probed mBERT and monolingual BERT models across six typologically diverse languages (German, English, Chinese, Russian, Finnish, Turkish) for lexical semantics. Their results show that pretrained monolingual BERT models encode significantly more lexical information than mBERT for a particular language, again hinting that mBERT cannot learn lexical information adequately for all of its 104 languages due to its limited model capacity.

Zhang et al. (2020) investigated the role of pretraining data size with RoBERTa; they found that the model already learns most syntactic and semantic features from pretraining on the corpora spanning 10M–100M word tokens, but still requires massive datasets to encode higher-level semantic and commonsense knowledge.

The work closest to ours is that of Rönnqvist et al. (2019). They compared mBERT to monolingual BERT models for six languages (German, English, Swedish, Danish, Norwegian, Finnish) on three different tasks. They find that mBERT lags behind its monolingual counterparts in terms of performance on cloze and generation tasks. They also identified clear differences among the six languages in terms of this performance gap. For example, the gap is smaller for German than for Finnish. Accordingly, they speculate that mBERT is undertrained with respect to individual languages. One shortcoming of their evaluation is that their set of tasks is limited, and their language sample is very narrow typologically; it remains unclear whether these findings extend to different language families and to structurally different tasks.

Despite recent efforts, a careful, systematic study within a *controlled* experimental setup, a diverse language sample and set of tasks is still lacking. We aim to address this gap in this work.

[1] https://bertlang.unibocconi.it/

3 Controlled Experimental Setup

We compare multilingual BERT with its monolingual counterparts in a spectrum of typologically diverse languages and across a variety of downstream tasks. By isolating and analyzing crucial factors contributing to downstream performance, such as used tokenizers and pretraining data, we can conduct unbiased and fair comparisons.

3.1 Language and Task Selection

The selection of languages has been guided by several (sometimes competing) criteria: **C1)** typological diversity; **C2)** availability of pretrained monolingual BERT models; **C3)** representation of the languages in standard evaluation benchmarks for a sufficient number of tasks.

Regarding C1, most high-resource languages belong to the same language families, thus sharing a majority of their linguistic features. Neglecting typological diversity inevitably leads to poor generalizability and the induction of biases (Gerz et al., 2018; Joshi et al., 2020; Ponti et al., 2019). Following recent work in multilingual NLP which pays particular attention to typological diversity (Clark et al., 2020; Hu et al., 2020; Ponti et al., 2020, *inter alia*), we experiment with a language sample covering a broad spectrum of language properties.

Regarding C2, for computational tractability, we only select languages with readily available BERT models. Unlike prior work, which typically lacks either language (Rönnqvist et al., 2019; Zhang et al., 2020) or task diversity (Wu and Dredze, 2020; Vulić et al., 2020), we ensure that our experimental framework takes both into account, thus also satisfying C3. Task diversity and generalizability is achieved in two ways. First, we select a combination of tasks driven by lower-level syntactic and higher-level semantic features (Lauscher et al., 2020b). Second, we also experiment with different task fine-tuning regimes, see later in §3.

Finally, we select a set of nine languages from eight different language families, as listed in Table 1.[2] [3] We evaluate mBERT and monolin-

[2] Note that, since we evaluate monolingual performance and not cross-lingual transfer performance, we require *training data* in the target language. Therefore, we are unable to leverage many of the available multilingual evaluation data such as XQuAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), or XNLI (Conneau et al., 2018). These evaluation sets do not provide any training portions for languages other than English.

[3] Additional details regarding our selection of languages and their pretrained models are available in Appendix A.1.

Language	ISO	Language Family	Pretrained BERT Model
Arabic	AR	Afroasiatic	AraBERT (Antoun et al., 2020)
English	EN	Indo-European	BERT (Devlin et al., 2019)
Finnish	FI	Uralic	FinBERT (Virtanen et al., 2019)
Indonesian	ID	Austronesian	IndoBERT (Wilie et al., 2020)
Japanese	JA	Japonic	Japanese-char BERT [17]
Korean	KO	Koreanic	KR-BERT (Lee et al., 2020)
Russian	RU	Indo-European	RuBERT (Kuratov and Arkhipov, 2019)
Turkish	TR	Turkic	BERTurk (Schweter, 2020)
Chinese	ZH	Sino-Tibetan	Chinese BERT (Devlin et al., 2019)

Table 1: Overview of selected languages and their respective pretrained monolingual BERT models.

gual BERT models on five downstream NLP tasks: named entity recognition (NER), sentiment analysis (SA), question answering (QA), universal dependency parsing (UDP), and part-of-speech tagging (POS).

Named Entity Recognition. We rely on the following NER datasets: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), FiNER (Ruokolainen et al., 2020), Chinese Literature (Xu et al., 2017), KMOU NER[29], WikiAnn (Pan et al., 2017; Rahimi et al., 2019).

Sentiment Analysis aims to classify the sentiment polarity (positive or negative). We rely on HARD (Elnagar et al., 2018), IMDb Movie Reviews (Maas et al., 2011), Indonesian Prosa (Purwarianti and Crisdayanti, 2019), Yahoo Movie Reviews[31], NSMC[32], RuReviews (Smetanin and Komarov, 2019), Turkish Movie and Product Reviews (Demirtas and Pechenizkiy, 2013), ChnSentiCorp[33].

Question Answering finds answers to questions within a context paragraph. We use SQuADv1.1 (Rajpurkar et al., 2016), KorQuAD 1.0 (Lim et al., 2019), SberQuAD (Efimov et al., 2020), TQuAD[36], DRCD (Shao et al., 2019), TyDiQA-GoldP (Clark et al., 2020).

Dependency Parsing predicts syntactic head-dependent relationships in a sentence. We rely on Universal Dependencies (Nivre et al., 2016, 2020) v2.6 (Zeman et al., 2020) for all languages.

Part-of-Speech Tagging classifies the corresponding part-of-speech tags for each word in a sentence. We again use Universal Dependencies (Nivre et al., 2016, 2020) v2.6 (Zeman et al., 2020).

Detailed descriptions of all the task data, including preprocessing steps, and which datasets are associated with which language, are provided in Appendix A.4.

3.2 Task-Based Fine-Tuning

Fine-Tuning Setup. We use the standard fine-tuning setup of Devlin et al. (2019) for all tasks besides UDP: for that, we use a transformer-based variant (Glavaš and Vulić, 2020) of the standard deep biaffine attention dependency parser (Dozat and Manning, 2017). Besides full model fine-tuning, we also evaluate all models within a more efficient setup based on adapters (Rebuffi et al., 2017; Houlsby et al., 2019; Stickland and Murray, 2019; Pfeiffer et al., 2020a,b,c,d; Lauscher et al., 2020a; Rücklé et al., 2020a,b, *inter alia*): additional parameter sets that are fine-tuned while the original pretrained model is kept frozen. Adapters have been shown to perform well for cross-lingual transfer by training language-specific adapters (Pfeiffer et al., 2020c,d), we here evaluate whether they perform equally well in monolingual setups.

In summary, we distinguish between four different setups for each task: **1)** fully fine-tune a monolingual BERT model; **2)** fully fine-tune mBERT on the task; **3)** inject a task adapter into mBERT, and fine-tune by updating the task adapter parameters (labeled $+A^{\text{Task}}$ henceforth); **4)** inject both a dedicated language adapter available via AdapterHub (Pfeiffer et al., 2020b), and a task adapter into mBERT, and then fine-tune by updating only the task adapter parameters ($+A^{\text{Lang, Task}}$).

For all settings, we average scores over three random initializations on the development set. On the test set, we report the results of the initialization that achieved the highest score on the development set.

Evaluation Measures. We report F_1 scores for NER, accuracy scores for SA and POS, unlabeled and labeled attachment scores (UAS & LAS) for UDP, and exact match and F_1 scores for QA.

Hyper-Parameters and Technical Details. We use Adam (Kingma and Ba, 2015) in all experiments, with initial learning rates of $3e-5$ for full fine-tuning, and $5e-4$ for the adapter-based setups, and linear learning rate decay.[4] During training, we evaluate a model every 500 gradient steps on the development set, saving the best-performing model.[5]

[4]These learning rates were fixed after running several preliminary experiments. Due to the large volume of our experiments, we were unable to tune all the hyper-parameters for each setting. We found that a higher learning rate works best for adapter-based fine-tuning since the task adapter parameters are learned from scratch (i.e., they are randomly initialized).

[5]Based on the respective evaluation measures. For QA and UDP, we use the F_1 scores and LAS, respectively.

We typically train for 10 epochs (full fine-tuning) or 30 epochs (adapter-based).[6] We rely on early stopping (Prechelt, 1998), terminating training if no performance gains are observed within five consecutive evaluation runs (= 2,500 steps). We train with batch size 32 and max sequence length 256 for all tasks except QA. In QA, the batch size is 24, max sequence length 384, query length 64, and document stride is set to 128.

3.3 Initial Results

We report our first set of results in Table 2. The results on development sets are available in the Appendix in Table 10. We find that the performance gap between monolingual models and mBERT does exist to a large extent, confirming the intuition from prior work. However, we also notice that the score differences are largely dependent on the language and task at hand. The largest performance gains of monolingual models over mBERT are found for FI, TR, KO, and AR. In contrast, mBERT outperforms the IndoBERT (ID) model in all tasks but SA, and performs competitively with the JA and ZH monolingual models on most datasets. In general, the gap is particularly narrow for POS tagging, where all models tend to score high (in most cases north of 95% accuracy). ID aside, we also see a clear trend for UDP, with monolingual models outperforming fully fine-tuned mBERT models, most notably for FI and TR, and fully fine-tuned mBERT models, in turn, outperforming the adapter-based models. In what follows, we seek to understand the causes of this behaviour in relation to different factors such as used tokenizers, corpora sizes, as well as languages and tasks in consideration.

In the remaining experiments, we focus on the full fine-tuning setup, as we find that fine-tuning mBERT with adapters appears to be effective only when the fully fine-tuned mBERT is also effective, compared to the monolingual models (e.g., in EN, ID, JA, ZH). Both adapter approaches ($+A^{\text{Lang, Task}}$, $+A^{\text{Task}}$) work similarly well. The former works best for UDP, whereas the latter works best for QA. Although language adapters are proven to yield significant gains over task-only adapters in cross-lingual settings (Pfeiffer et al., 2020c,d), we believe that either choice is generally suitable, albeit not ideal if maximum performance is required, to tackle monolingual tasks.[7]

[6]The exceptions are FI and ID QA: there we do full fine-tuning for 20 epochs due to slower convergence.

[7]We further elaborate on this verdict in Appendix B.3.

Lang	Model	NER	SA	QA	UDP	POS
		Test F_1	Test Acc	Dev EM / F_1	Test UAS / LAS	Test Acc
AR	Monolingual	**91.1**	**95.9**	**68.3 / 82.4**	**90.1 / 85.6**	96.8
	mBERT	90.0	95.4	66.1 / 80.6	88.8 / 83.8	96.8
	+ $A^{\text{Lang, Task}}$	89.7	95.7	66.9 / 81.0	88.0 / 82.8	**96.8**
	+ A^{Task}	89.6	95.6	66.7 / 81.1	87.8 / 82.6	96.8
EN	Monolingual	**91.5**	**91.6**	80.5 / 88.0	**92.1 / 89.7**	**97.0**
	mBERT	91.2	89.8	**80.9 / 88.4**	91.6 / 89.1	96.9
	+ $A^{\text{Lang, Task}}$	91.4	89.4	80.1 / 87.7	91.3 / 88.7	96.7
	+ A^{Task}	90.5	89.8	79.9 / 87.6	91.0 / 88.3	96.7
FI	Monolingual	**92.0**	—	**69.9 / 81.6**	**95.9 / 94.4**	**98.4**
	mBERT	88.2	—	66.6 / 77.6	91.9 / 88.7	96.2
	+ $A^{\text{Lang, Task}}$	88.4	—	65.7 / 77.1	91.8 / 88.5	96.6
	+ A^{Task}	88.5	—	65.2 / 77.3	90.8 / 87.0	95.7
ID	Monolingual	91.0	**96.0**	66.8 / 78.1	85.3 / 78.1	92.1
	mBERT	**93.5**	91.4	**71.2** / 82.1	**85.9 / 79.3**	**93.5**
	+ $A^{\text{Lang, Task}}$	93.5	93.6	70.8 / 82.2	85.4 / 78.1	93.4
	+ A^{Task}	93.5	90.6	70.6 / **82.5**	84.8 / 77.4	93.4
JA	Monolingual	72.4	88.0	— / —	**94.7 / 93.0**	**98.1**
	mBERT	**73.4**	87.8	— / —	94.0 / 92.3	97.8
	+ $A^{\text{Lang, Task}}$	70.9	88.4	— / —	93.5 / 91.6	97.8
	+ A^{Task}	71.5	**88.6**	— / —	93.6 / 91.6	97.7
KO	Monolingual	**88.8**	**89.7**	**74.2 / 91.1**	**90.3 / 87.2**	**97.0**
	mBERT	86.6	86.7	69.7 / 89.5	89.2 / 85.7	96.0
	+ $A^{\text{Lang, Task}}$	86.2	86.3	70.0 / 89.8	88.3 / 84.3	96.2
	+ A^{Task}	86.2	86.5	69.8 / 89.7	87.8 / 83.9	96.2
RU	Monolingual	**91.0**	**95.2**	**64.3 / 83.7**	**93.1 / 89.9**	**98.4**
	mBERT	90.0	95.0	63.3 / 82.6	91.9 / 88.5	98.2
	+ $A^{\text{Lang, Task}}$	89.0	94.7	62.8 / 82.4	91.8 / 88.1	98.2
	+ A^{Task}	89.6	94.7	62.9 / 82.5	92.0 / 88.3	98.2
TR	Monolingual	92.8	**88.8**	**60.6 / 78.1**	**79.8 / 73.2**	**96.9**
	mBERT	**93.8**	86.4	57.9 / 76.4	74.5 / 67.4	95.7
	+ $A^{\text{Lang, Task}}$	93.5	84.8	56.9 / 75.8	73.0 / 64.7	95.9
	+ A^{Task}	93.0	83.9	55.3 / 75.1	72.4 / 64.1	95.7
ZH	Monolingual	**76.5**	**95.3**	82.3 / 89.3	**88.6 / 85.6**	**97.2**
	mBERT	76.1	93.8	82.0 / 89.3	88.1 / 85.0	96.7
	+ $A^{\text{Lang, Task}}$	75.4	94.8	82.1 / 89.4	87.3 / 83.8	96.4
	+ A^{Task}	75.2	94.1	**82.4 / 89.6**	87.5 / 83.9	96.5

Table 2: Model Performances on Named Entity Recognition (NER), Sentiment Analysis (SA), Question Answering (QA), Universal Dependency Parsing (UDP, and Part-of-Speech Tagging (POS). We use development (dev) sets only for QA. Finnish (FI) SA and Japanese (JA) QA lack respective datasets.

4 Tokenizer vs. Corpus Size

4.1 Pretraining Corpus Size

The size of the pretraining corpora plays an important role in the performance of transformers (Liu et al., 2019; Conneau et al., 2020; Zhang et al., 2020, *inter alia*). Therefore, we compare how much data each monolingual model was trained on with the amount of data in the respective language that mBERT has seen during training. Given that mBERT was trained on entire Wikipedia dumps[8], we estimate the latter by the total number of words

[8]https://github.com/google-research/bert/blob/master/multilingual.md

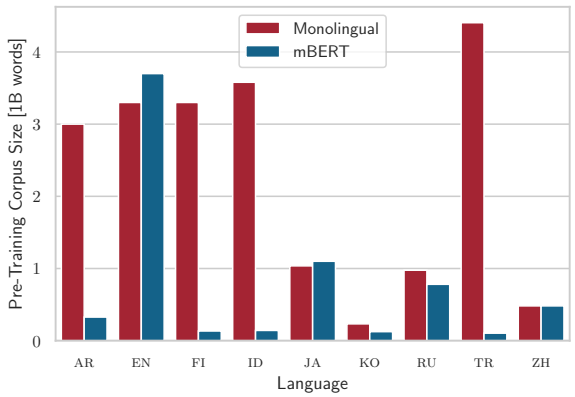


Figure 1: The number of words (in Billions) in monolingual pretraining corpora versus the respective monolingual portions of mBERT's pretraining corpus

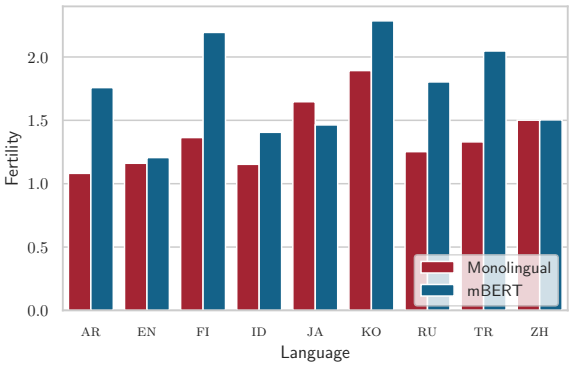


Figure 2: Subword fertility (i.e., the average number of subwords produced per tokenized word (Ács, 2019)) of monolingual tokenizers versus the mBERT tokenizer.

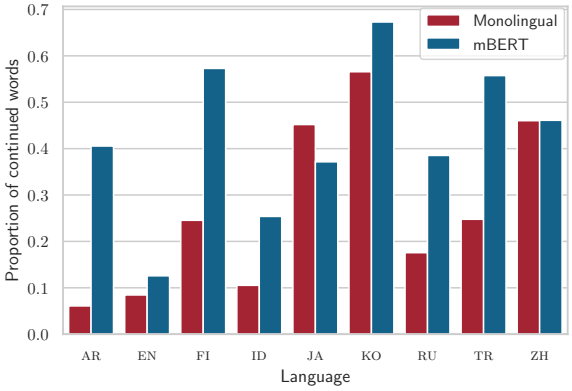


Figure 3: Proportion of continued words (i.e., words split into multiple subword tokens (Ács, 2019)) in monolingual corpora tokenized by monolingual models vs. mBERT.

across all articles listed for each Wiki[9]. For the monolingual models, we extract information on pretraining data from the model documentation. If no exact numbers are explicitly stated, and the pretraining corpora are unavailable to us, we make estimations based on the information the authors provide. Details on any particular assumptions and estimations we make are given in Appendix A.2. Our findings are depicted in Figure 1. For EN, JA, RU, and ZH, both the respective monolingual model and mBERT were trained on similar amounts of monolingual data. On the other hand, we see that the AR, ID, FI, KO, and TR monolingual models were trained from about twice (KO) up to more than 40 times (TR) as much data in their language than mBERT.

4.2 Tokenizer

Compared to monolingual models, mBERT is substantially more limited in terms of the "space", that is, the parameter budget it can allocate for each of its 104 languages in its vocabulary. Additionally, monolingual tokenizers are typically trained by native speaking experts aware of relevant linguistic phenomena exhibited by their target language. We thus inspect how this affects the tokenizations of monolingual data produced by our sample of monolingual models and mBERT. We tokenize examples from Universal Dependencies (Nivre et al., 2016, 2020) v2.6 (Zeman et al., 2020) treebanks (further details given in Appendix A.3) and compute two metrics (Ács, 2019). The first metric is the subword fertility, measuring the average number of subwords produced per tokenized word. A minimum fertility value of 1 means that the tokenizer's vocabulary contains every single word in the tokenized text. We plot the fertility scores in Table 2. We find that mBERT has similar fertility values as its monolingual counterparts for EN, ID, JA, and ZH. In contrast, mBERT has a much higher fertility for AR, FI, KO, RU, and TR, indicating that such languages may be over-segmented. mBERT's fertility is the lowest for EN, which is likely due to mBERT having seen the most data in this language during training, as well as due to English being a morphologically poor language in comparison to languages such as AR, FI, RU, or TR. The JA model is the only monolingual one with a fertility higher than mBERT because the JA tokenizer is character-based and thus by design produces a maximum number of word pieces.

The second metric is the proportion of words

[9]Based on numbers from https://meta.m.wikimedia.org/wiki/List_of_Wikipedias

in the corpus where the tokenized word is continued across at least two sub-tokens (denoted by continuation symbols ##). Whereas the fertility is concerned with how aggressively a tokenizer splits, the proportion of these *continued words* measures how often it splits words. Intuitively, low scores are preferable for both metrics as they indicate that the tokenizer is well suited to the language. The plots in Figure 3 show similar trends as with the fertility statistic. In addition to AR, FI, KO, RU, and TR, where there were already conspicuous differences in fertility, mBERT also produces a proportion of continued words more than twice as high as the monolingual model for ID.

We discuss additional tokenization statistics, further highlighting the differences (or lack thereof) between the individual monolingual tokenizers and the mBERT tokenizer, in Appendix B.1.

4.3 New Pretrained Models

The differences in pretraining corpora and tokenizer statistics from the previous sections seem to align with the variations in downstream performance across languages. In particular, it appears that the performance gains of monolingual models over mBERT are larger for languages where the differences between the respective tokenizers and pretraining corpora sizes are also larger (AR, FI, KO, RU, TR) and vice-versa (EN, JA, ZH).[10] Therefore, we hypothesize that both the data size and the tokenizer are among the main driving forces of downstream task performance. In order to disentangle the effects of these two factors, we pretrain new models for AR, FI, ID, KO, and TR (the languages that exhibited the largest discrepancies regarding the two factors) on Wikipedia data.

We train three model variants for each language. First, we train two new monolingual BERT models on the same data, one with the original monolingual tokenizer (`wiki-mono-mono`) and one with the mBERT tokenizer (`wiki-mono-mBERT`).[11] Additionally, we retrain the embedding layer of mBERT with the respective monolingual tokenizer (`wiki-mBERT-retrained`). Having the `wiki-mono-mono` models to compare against the monolingual models trained on significantly more data but with the same tokenizer, we implicitly disentangle the effect of the data size.

Pretraining Setup. We pretrain new BERT models for each language on its respective Wikipedia dump.[12] We apply two preprocessing steps to obtain clean data for pretraining. First, we use WikiExtractor (Attardi, 2015) to extract text passages from the raw dumps. Next, we follow Pyysalo et al. (2020) and utilize UDPipe (Straka et al., 2016) parsers pretrained on UD data[13] to segment the extracted text passages into texts with document, sentence, and word boundaries.

Following Wu and Dredze (2020), we only use masked language modeling (MLM) as pretraining objective and omit the next sentence prediction task as Liu et al. (2019) find it does not yield performance gains. We otherwise mostly follow the default pretraining procedure by Devlin et al. (2019).

We pretrain the new monolingual models (`wiki-mono`) from scratch for 1M steps with batch size 64. We choose a sequence length of 128 for the first 900,000 steps and 512 for the remaining 100,000 steps. We enable whole word masking (Devlin et al., 2019) for the FI monolingual models, following the pretraining procedure for FinBERT (Virtanen et al., 2019). For the retrained mBERT models, we run MLMing for 250,000 steps (similar to Artetxe et al. (2020)) with batch size 64 and sequence length 512, otherwise using the same hyper-parameters as for the monolingual models. We freeze all parameters outside the embedding layer. For more details see Appendix A.5.

Results. We perform the same evaluations on downstream tasks for our new models as described in §3. We report the results in Table 3. Full results including development set scores are available in Table 11 in the appendix.

Our results show that the models trained with dedicated monolingual tokenizers outperform their counterparts with multilingual tokenizers in most tasks, with particular consistency for QA, UDP, and SA. In NER, the models trained with multilingual tokenizers score competitively or higher than the monolingual ones in half of the cases. Overall, the

[10] The only exception is ID, where the monolingual model has seen significantly more data and also scores lower on the tokenizer metrics, yet underperforms mBERT in most tasks. We suspect this exception to be due to the IndoBERT model being uncased, whereas the remaining models are cased.

[11] The only exception is ID, where, instead of relying on the uncased IndoBERT tokenizer by Wilie et al. (2020), we introduce a new *cased* tokenizer with identical vocabulary size (30,521).

[12] We use Wiki dumps from June 20, 2020 - e.g. fiwiki-20200720-pages-articles.xml.bz2 for FI.

[13] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3131

Lang	Model	Tokenizer	NER	SA	QA	UDP	POS
			Test F_1	Test Acc	Dev EM / F_1	Test UAS / LAS	Test Acc
AR	wiki-mono	mono	**91.7**	**95.6**	**67.7 / 81.6**	**89.2 / 84.4**	96.6
	wiki-mono	mBERT	90.0	95.5	64.1 / 79.4	88.8 / 84.0	**97.0**
	mBERT	mono	**91.2**	**95.4**	**66.9 / 81.8**	**89.3 / 84.5**	96.4
	mBERT	mBERT	90.0	95.4	66.1 / 80.6	88.8 / 83.8	**96.8**
FI	wiki-mono	mono	89.1	—	**66.9 / 79.5**	**93.7 / 91.5**	**97.3**
	wiki-mono	mBERT	**90.0**	—	65.1 / 77.0	93.6 / 91.5	97.0
	mBERT	mono	88.1	—	66.4 / **78.3**	**92.4 / 89.6**	**96.6**
	mBERT	mBERT	**88.2**	—	**66.6** / 77.6	91.9 / 88.7	96.2
ID	wiki-mono	mono	92.5	**96.0**	**73.1 / 83.6**	**85.0** / 78.5	**93.9**
	wiki-mono	mBERT	**93.2**	94.8	67.0 / 79.2	84.9 / **78.6**	93.6
	mBERT	mono	**93.9**	**94.6**	**74.1 / 83.8**	**86.4 / 80.2**	**93.8**
	mBERT	mBERT	93.5	91.4	71.2 / 82.1	85.9 / 79.3	93.5
KO	wiki-mono	mono	**87.1**	**88.8**	**72.8 / 90.3**	**89.8 / 86.6**	**96.7**
	wiki-mono	mBERT	85.8	87.2	68.9 / 88.7	88.9 / 85.6	96.4
	mBERT	mono	86.6	**88.1**	**72.9 / 90.2**	**90.1 / 87.0**	**96.5**
	mBERT	mBERT	**86.6**	86.7	69.7 / 89.5	89.2 / 85.7	96.0
TR	wiki-mono	mono	**93.4**	**87.0**	**56.2 / 73.7**	**76.1 / 68.9**	96.3
	wiki-mono	mBERT	93.3	84.8	55.3 / 72.5	75.3 / 68.3	**96.5**
	mBERT	mono	93.7	85.3	**59.4 / 76.7**	**77.1 / 70.2**	**96.3**
	mBERT	mBERT	**93.8**	**86.4**	57.9 / 76.4	74.5 / 67.4	95.7

Table 3: Performances of our new `wiki-mono` and `wiki-mbert-retrained` models fine-tuned for Named Entity Recognition (NER), Sentiment Analysis (SA), Question Answering (QA), Universal Dependency Parsing (UDP), and Part-of-Speech Tagging (POS). We add the original fully fine-tuned mBERT and group model counterparts w.r.t. tokenizer choice to facilitate a direct comparison between respective counterparts. mBERT model with mono tokenizer refers to `wiki-mbert-retrained` and mBERT model with mBERT tokenizer refers to the original fully fine-tuned mBERT. We use development sets only for QA.

performance gap is the smallest for POS tagging (at most 0.5% accuracy). We observe the largest gaps for QA (6.1 EM / 4.4 F_1 in ID), SA (3.2% accuracy in ID), and UDP (2.8 LAS in TR).

Overall we find that for 39 out of 48 task, model, and language combinations, the monolingual tokenizer outperforms the mBERT version. We were able to improve the monolingual performance of mBERT for 19 out of 24 languages and tasks by only replacing the tokenizer and, thus, leveraging a specialized monolingual version. These results establish that, in fact, the tokenizer plays a fundamental role in the downstream task performance.

5 Further Analysis

5.1 Qualitative Analysis

Qualitatively and at first glance, our results displayed in Table 2 seem to confirm the prevailing view that monolingual models are more effective than multilingual models (Rönnqvist et al., 2019; Antoun et al., 2020; de Vries et al., 2019; Virtanen et al., 2019, *inter alia*). However, our broad range of experiments reveals certain nuances that were previously undiscovered.

In contrast to previous work which primarily attributes gaps in performance to mBERT being under-trained with respect to individual languages (Rönnqvist et al., 2019; Wu and Dredze, 2020), our results, when disentangling the effect of the tokenizer (as seen in Table 3), convincingly show that a large portion of existing performance gaps can be attributed to the capability of the designated tokenizer. When choosing a monolingual tokenizer that scores significantly lower in fertility and the proportion of continued words than the mBERT tokenizer (such as for AR, FI, ID, KO, TR), performance gains can be made relatively consistently, irrespective of whether the models themselves are monolingual (`wiki-mono-mono` versus `wiki-mono-mbert`) or multilingual (`wiki-mbert-retrained` versus fully fine-tuned mBERT).

Whenever the differences between monolingual models and mBERT with respect to the tokenizer (as measured by the fertility or proportion of continued words) and the pretraining corpus size are small, such as for EN, JA, and ZH, the performance gap is typically also small. In QA, we even find mBERT to be favorable for these languages. Therefore, we conclude that monolingual models are not superior to multilingual models per se, but rather most of the time, gain an unfair advantage in a direct comparison by incorporating more pretraining data and using more capable tokenizers.

Note that similar findings are observed with both modes of task fine-tuning. We discuss the effectiveness of adapter-based fine-tuning in the context of monolingual tasks in Appendix B.3.

5.2 Correlation Analysis

To uncover some of the hidden patterns in our results (Tables 2 and 3), we perform a statistical analysis assessing the correlation between the individual factors (pretraining data size, subword fertility, proportion of continued words) and the downstream performance. Although our framework may not provide enough data points to be statistically representative, we argue that the correlation coefficient can still provide reasonable indications and reveal patterns in our results not immediately evident by looking at the tables.

Figure 4 shows that both decreases in the proportion of continued words and the fertility correlate with an increase in downstream performance relative to fully fine-tuned mBERT across all tasks.

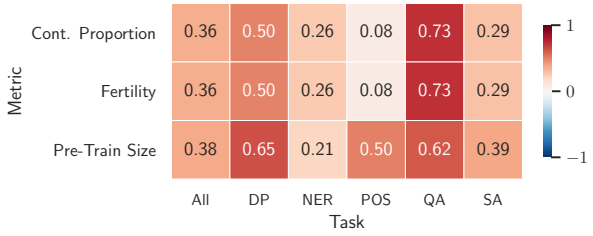


Figure 4: Spearman's ρ correlation of a relative decrease in the proportion of continued words (Cont. Proportion), a relative decrease in fertility, and a relative increase in pretraining corpus size with a relative increase in downstream performance over fully fine-tuned mBERT. For the proportion of continued words and the fertility, we consider fully fine-tuned mBERT, the `wiki-mono` models, and the `wiki-mbert-retrained` models. For the pretraining corpus size, we consider the original monolingual models and the `wiki-mono-mono` models. We exclude the ID models as explained in Appendix B.2.

The correlation is stronger for UDP and QA, where we find models with monolingual tokenizers to outperform their counterparts with the mBERT tokenizer consistently. The correlation is weaker for NER and POS tagging, which is also expected, considering the inconsistency of the results.[14]

Overall, we find that the fertility and the proportion of continued words have a similar effect on the monolingual downstream performance as the corpus size for pretraining. This indicates that the tokenizer's capability of representing a language plays an equal role as the amount of data a model sees during training. Consequently, choosing a suboptimal tokenizer will result in a deterioration of the downstream performance.

6 Conclusion

In this work, we have conducted the first widely targeted empirical investigation concerning the monolingual performance of monolingual and multilingual models. While our results support the existence of a performance gap in most but not all languages and tasks, further analysis revealed that this performance gap is often significantly smaller than previously assumed and only exacerbated in certain languages by incorporating substantially more pretraining data and using more capable, monolingual tokenizers.

Further, we have disentangled the effect of the pretrained corpus size from the tokenizers, in order to identify the importance of either on the downstream task performance. We have trained new monolingual models on the same data but with two different tokenizers; one being the dedicated tokenizer of the monolingual model provided by native speakers; the other being the automatically generated multilingual mBERT tokenizer. We find that for (almost) every task and language, the monolingual tokenizer outperforms the mBERT tokenizer, establishing that a specialized vocabulary plays an equally important role on the downstream performance as the pretraining data set size.

Consequently, our results suggest that a more deliberate balancing of individual languages' representations within the tokenizer's vocabulary (e.g., by merging monolingual vocabularies) can close the gap between monolingual and multilingual models in cases where the tokenizer currently makes the difference.

[14]For further information see Appendix B.2.

Acknowledgments

Jonas Pfeiffer is supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY center. The work of Ivan Vulić is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909).

We thank Nils Reimers and Prasetya Ajie Utama for insightful feedback and suggestions on a draft of this paper.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

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

Giuseppe Attardi. 2015. Wikiextractor. *GitHub Repository*.

Enkhbold Bataa and Joshua Wu. 2019. An investigation of transfer learning-based sentiment analysis in Japanese. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4652–4657, Florence, Italy. Association for Computational Linguistics.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual bert, a small treebank, and a small corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1324–1334.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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

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

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Erkin Demirtas and Mykola Pechenizkiy. 2013. Cross-lingual polarity detection with machine translation. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '13, New York, NY, USA. Association for Computing Machinery.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France. OpenReview.net.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. Sberquad – russian reading comprehension dataset: Description and analysis. *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 3–15.

Ibrahim Abu El-khair. 2016. 1.5 billion words arabic corpus. *arXiv preprint*.

Ashraf Elnagar, Yasmin S. Khalifa, and Anas Einea. 2018. *Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications*, pages 35–52. Springer International Publishing, Cham, Switzerland.

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327, Brussels, Belgium. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2020. Is supervised syntactic parsing beneficial for language understanding? an empirical investigation. *arXiv preprint*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, CA, USA. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421, Virtual. PMLR.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2019"*, pages 333–339, Moscow, Russia.

Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavas. 2020a. Common Sense or World Knowledge? Investigating Adapter-Based Knowledge Injection into Pretrained Transformers. *arXiv preprint*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020b. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online.

Sangah Lee, Hansol Jang, Yunmee Baik, Suzi Park, and Hyopil Shin. 2020. Kr-bert: A small-scale korean-specific language model. *arXiv preprint*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean qa dataset for machine reading comprehension. *arXiv preprint*.

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. *arXiv preprint*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA. OpenReview.net.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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 *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [mask]? making sense of language-specific bert models. *arXiv preprint*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 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)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. *arXiv preprint*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020d. UNKs Everywhere: Adapting Multilingual Language Models to New Scripts. *arXiv preprint*.

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

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, volume 2481. CEUR.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 55–69, Berlin, Germany. Springer-Verlag.

A. Purwarianti and I. A. P. A. Crisdayanti. 2019. Improving bi-lstm performance for indonesian sentiment analysis using paragraph vector. In *Proceedings of the 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–5, Yogyakarta, Indonesia. IEEE.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2020. Wikibert models: deep transfer learning for many languages. *arXiv preprint*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30, pages 506–516, Long Beach, CA, USA. Curran Associates, Inc.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020a. AdapterDrop: On the Efficiency of Adapters in Transformers. *arXiv preprint*.

Andreas Rücklé, Jonas Pfeiffer, and Iryna Gurevych. 2020b. MultiCQA: Zero-shot transfer of self-supervised text matching models on a massive scale. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2471–2486, Online. Association for Computational Linguistics.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2020. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, 54(1):247–272.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual bert fluent in language generation? In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.

Stefan Schweter. 2020. Berturk - bert models for turkish. Zenodo.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2019. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint*.

Sergey Smetanin and Michail Komarov. 2019. Sentiment analysis of product reviews in russian using convolutional neural networks. In *Proceedings of the 2019 IEEE 21st Conference on Business Informatics (CBI)*, volume 01, pages 482–486, Moscow, Russia. IEEE.

Asa Cooper Stickland and Iain Murray. 2019. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995, Long Beach, California, USA. PMLR.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-u files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Long Beach, CA, USA. Curran Associates, Inc.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A dutch bert model. *arXiv preprint*.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

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

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, Lukasz 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. *arxiv preprint*.

Jingjing Xu, Ji Wen, Xu Sun, and Qi Su. 2017. A discourse-level named entity recognition and relation extraction dataset for chinese literature text. *arXiv preprint*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint*.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Angelina Aquino, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, 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, Ethan Chi, 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 Ilar-

raza, 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, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Hildur Jónsdóttir, 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, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, 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 H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, 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, Hiroshi Matsuda, 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 Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Arzucan Özgür, Balkız Öztürk Başaran, 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, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Salvatore Scarlata, 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, Maria Skachedubova, 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, Samson Tella, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, 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, Aya Wakasa, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Hanzhi Zhu, and Anna Zhuravleva. 2020. Universal dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2020. When do you need billions of words of pretraining data? *arXiv preprint*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015*

IEEE International Conference on Computer Vision (ICCV), pages 19–27, Santiago, Chile. IEEE Computer Society.

Judit Ács. 2019. Exploring bert's vocabulary. *Blog Post*.

A Reproducibility

A.1 Selection of Pretrained Models

All of the pretrained models we use, besides the Korean KR-BERT[15], are openly available on the HuggingFace model hub[16] and compatible with the HuggingFace transformers Python library (Wolf et al., 2020). Models and their associated tokenizers have unique identifiers by which they can be downloaded and integrated into Python code.

Whenever both uncased and cased model variants are available, we select the cased variant because the mBERT model (`bert-base-multilingual-cased`) is also cased. Furthermore, based on initial evaluations, we find that using cased models is generally advantageous for our task sample. Accordingly, we select the cased BERT (`bert-base-cased`) model (Devlin et al., 2019) for EN, the cased FinBERT (`TurkuNLP/bert-base-finnish-cased-v1`) model (Virtanen et al., 2019) for FI, and the cased BERTurk (`dbmdz/bert-base-turkish-cased`) model (Schweter, 2020) for TR.

For AR, Antoun et al. (2020) provide two versions: AraBERTv0.1 and AraBERTv1. The only difference between them is that the pretraining data for version 1 was pre-segmented with Farasa (Abdelali et al., 2016) before applying WordPiece. Based on their evaluations, as well as our own preliminary experiments, both AraBERT models perform similarly well, each one being the superior choice for certain tasks. Therefore, we select v0.1, which does not require pre-segmentation in the fine-tuning stage.

For JA, the publishers from Inui Laboratory, Tohoku University, provide four models[17]: character-tokenized (with and without whole word masking enabled) and subword tokenized (with and without whole word masking enabled). We select the character-tokenized Japanese BERT model because it achieved considerably higher scores on preliminary NER fine-tuning evaluations. Since mBERT was trained without WWM, we also select the Japanese model trained without WWM.

For KO, we use the KR-BERT (Lee et al., 2020) model[18] by the Computational Linguis-

[15] Which is openly available at `https://github.com/snunlp/KR-BERT`

[16] `https://huggingface.co/models`

[17] `https://github.com/cl-tohoku/bert-japanese`

[18] `https://github.com/snunlp/KR-BERT`

tics Lab at Seoul National University. We have also experimented with the KoBERT[19] model (monologg/kobert), introduced by SKTBrain, but found that it exhibited significant performance issues in QA[20]. The KR-BERT model in particular offers a more suitable tokenizer than KoBERT and outperforms KoBERT on several Korean benchmark datasets despite having been trained on less data (Lee et al., 2020).

The ID IndoBERT (Wilie et al., 2020) model (indobenchmark/indobert-base-p2), the RU RuBERT (Kuratov and Arkhipov, 2019) model (DeepPavlov/rubert-base-cased) by DeepPavlov , and the ZH Chinese-BERT (Devlin et al., 2019) model (bert-base-chinese) by Google are, to the best of our knowledge, the only openly available models for their respective language following the original *bert-base* architecture and pretraining procedure by Devlin et al. (2019).

A.2 Estimating the Pretraining Corpora Sizes

For reference, mBERT was pretrained on the entire Wikipedia dumps of all languages it covers (Devlin et al., 2019).[21] All of the monolingual models were also pretrained on their respective monolingual Wikipedia dumps. However, most publishers employed additional pretraining data from other sources for their monolingual models (Antoun et al., 2020; Virtanen et al., 2019; Kuratov and Arkhipov, 2019; Devlin et al., 2019; Lee et al., 2020).

AraBERT (Antoun et al., 2020) was additionally pretrained on manually scraped Arabic news articles, the Open Source International Arabic News (OSIAN) Corpus (Zeroual et al., 2019), and the 1.5B words Arabic Corpus (El-khair, 2016), for a total of about 3.3B words. Devlin et al. (2019) have also included the BooksCorpus (Zhu et al., 2015) for BERT, for a total of about 3.3B words. FinBERT (Virtanen et al., 2019) was also pretrained on aggressively cleaned and filtered data from multiple Finnish news corpora, online discussion posts, and an unrestricted internet crawl, for a total of about 3B words. KR-BERT (Lee et al., 2020) was trained on 233M words (20M sentences), including both Korean Wikipedia and news data. BERTurk (Schweter, 2020) was, in addition to Wikipedia, pretrained on the Turkish OSCAR corpus (Ortiz Suárez et al., 2020), a special corpus by Kemal Oflazer[22], and various OPUS[23] corpora for a total of about 4.4B words. IndoBERT (Wilie et al., 2020) was trained on a ∼3.6B word corpus collected from 15 different sources, which also include the Indonesian OSCAR corpus (Ortiz Suárez et al., 2020), an Indonesian Wiki dump, and data from Twitter. The monolingual pretraining corpus for RuBERT (Kuratov and Arkhipov, 2019) also included Russian news articles, accounting for about 20% of the total data. Kuratov and Arkhipov (2019) only further state on GitHub[24] that they used about 6.5GB of data in total. Based on the number of words in the Russian Wikipedia dump (781M on September 10, 2020, according to Wikimedia[25]), and the insight that the Wikipedia dump constituted 80% of the total corpus, we estimate the RuBERT pretraining corpus at about 976M words. Considering that Kuratov and Arkhipov (2019) most likely filtered the data and used an older Wikipedia dump, this estimation should serve as an upper bound.

The JA[26] and ZH (Devlin et al., 2019) BERT models were only pretrained on Wikipedia data, so there should not be any major differences to mBERT in terms of training corpus size. For ZH, we can assume that Devlin et al. (2019) used the same data cleaning and filtering procedure as for mBERT, and, according to a GitHub issue[27], the ZH corpus consisted of about 25M sentences. We do not know the exact number of words, so we make an estimation based on the number of words in the raw Wikipedia dump (482M on September 10, 2020, according to Wikimedia[25]). For JA, we use the publishers' scripts to recreate the corpus and calculate a total word count of about 1.03B. We use a newer Wikipedia dump[28] because the one used by the publishers is not available anymore.

We estimate the language-specific shares of the mBERT pretraining corpus by word counts of the

[19] https://github.com/SKTBrain/KoBERT

[20] in line with results by https://github.com/monologg/KoBERT-KorQuAD

[21] https://github.com/google-research/bert/blob/master/multilingual.md#list-of-languages

[22] http://www.andrew.cmu.edu/user/ko/

[23] http://opus.nlpl.eu/

[24] https://github.com/deepmipt/DeepPavlov/issues/1074

[25] https://meta.m.wikimedia.org/wiki/List_of_Wikipedias

[26] https://github.com/cl-tohoku/bert-japanese

[27] https://github.com/google-research/bert/issues/155

[28] jawiki-20200820-pages-articles-multistream.xml.bz2

respective raw Wikipedia dumps, according to numbers obtained from Wikimedia[25] on September 10, 2020 (December 10, 2020 for ID and TR). We obtain the following numbers:

- 327M words for AR
- 3.7B for EN
- 134M for FI
- 142M for ID
- 1.1B for JA
- 125M for KO
- 781M for RU
- 104M for TR
- 482M for ZH

Devlin et al. (2019) only included text passages from the articles, and used older Wikipedia dumps, so these numbers should serve as as upper limits, yet be reasonably accurate.

A.3 Data for Tokenizer Analyses

We tokenize the training and development splits of the UD (Nivre et al., 2016, 2020) v2.6 (Zeman et al., 2020) treebanks listed in Table 4.

A.4 Fine-Tuning Datasets

Named Entity Recognition

We use the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) for EN, the FiNER dataset (Ruokolainen et al., 2020) for FI, the Chinese Literature Dataset (Xu et al., 2017) for ZH, and the Korea Maritime and Ocean University (KMOU) NER dataset[29] for KO. For AR, ID, JA, RU, and TR, we use the respective portions of the WikiAnn dataset, which was originally introduced by Pan et al. (2017) to provide NER data for 282 languages and further balanced and split for 41 of these languages by Rahimi et al. (2019).

Except for the KMOU dataset, all NER datasets are readily split into training, development, and test splits. The KMOU dataset consists of hundreds of single files. Similar to the KoBERT NER implementation[30], we use the `EXOBRAIN_NE_CORPUS_009.txt` file as our dev set, the `EXOBRAIN_NE_CORPUS_010.txt` as our test set, and the rest of the files for training.

We use simple shell-based preprocessing to transform all datasets into the same format, but do not perform any additional cleaning of the data. In a final preprocessing step, we split sequences larger than the model's specified maximum sequence length. Table 5 provides an overview of the NER datasets, including the number of sequences per dataset split after this final preprocessing step.

Sentiment Analysis

We try to select datasets for which reference scores are available, whenever possible. Furthermore, we ensure that the datasets are balanced, meaning that there are equally many positive and negative instances. By doing so, we prevent the model from potentially learning polarity biases present in the data. We perform additional preprocessing only when necessary, but ensure that all datasets are in the same format and are split three-way for training, development, and testing.

We select the HARD dataset (Elnagar et al., 2018) for AR, the IMDb movie reviews dataset (Maas et al., 2011) for EN, the prosa sentiment analysis dataset (Purwarianti and Crisdayanti, 2019) for ID, the Yahoo Movie Reviews datasets[31] for JA, the Naver Sentiment Movie Corpus (NSMC)[32] for KO, the RuReviews dataset (Smetanin and Komarov, 2019) for RU, the movie and product reviews datasets by Demirtas and Pechenizkiy (2013), merged into a single corpus, for TR, and the ChnSentiCorp dataset[33] for ZH. To the best of our knowledge, an openly available SA dataset for FI currently does not exist.

For the HARD (Elnagar et al., 2018) dataset, we shuffle and split the `balanced-reviews.tsv` file using 80% of the data for training, and 10% each for development and testing (80/10/10 split). The dataset contains four classes - very negative (1), negative (2), positive (4), very positive (5), so we combine classes (1) and (2) into a single negative polarity (0), and (4) and (5) into a single positive polarity (1). Finally, we clean the texts by removing line breaks and replacing tabs by single spaces.

The ChnSentiCorp[33] dataset, which is also used for evaluation by Cui et al. (2020), is already split and we do not perform any additional preprocessing.

The EN IMDb movie reviews (Maas et al., 2011) dataset comes as a collection of single text files separated into two equally large sets for training and

[29] https://github.com/kmounlp/NER

[30] https://github.com/eagle705/pytorch-bert-crf-ner#ner-tagset

[31] https://github.com/dennybritz/sentiment-analysis/tree/master/data

[32] https://www.lucypark.kr/docs/2015-pyconkr/#39

[33] https://github.com/pengming617/bert_classification/tree/master/data

testing, each further separated into equally large sets of positive and negative instances. We randomly choose 20% of the training instances for our development set, keeping the test set the same to facilitate comparisons with models in the literature. Finally, we combine all small files into one large file per data split and clean the data in the same way as for the HARD dataset.

We obtain the ID prosa dataset (Purwarianti and Crisdayanti, 2019) from Kaggle.[34] We do not use the version of the dataset by Wilie et al. (2020) because it is not binary, not balanced, and the test set is hidden. Instead, we obtain the original dataset, remove neutral sentiment instances, and balance the dataset. We split the original training data 90/10 for training and development, and use the original testing data for testing.

The JA Yahoo Movie Reviews dataset[31] is also used by Bataa and Wu (2019). The dataset contains 5 classes (ratings from 1 to 5) and is unbalanced. To obtain a binary and balanced dataset, we discard neutral instances (3), combine negative and positive instances as we do for the HARD dataset, and randomly discard instances from the dominant polarity until both polarities have equal support. Finally, we use an 80/10/10 split and remove line breaks from the examples. We end up with the same number of instances per split as Bataa and Wu (2019).

The KO Naver Sentiment Movie Corpus (NSMC[32]) is a balanced dataset based on positive and negative reviews from the Naver Movies website. It is also used for evaluation by the publishers of KR-BERT (Lee et al., 2020) and KoBERT[19]. It comes in a 75/25 split for training/testing. We randomly select of 20% of the training data for development as we do for the English IMDb dataset, but do not perform any additional preprocessing.

The RU RuReviews dataset (Smetanin and Komarov, 2019) was collected from product reviews for women's clothing and accessories. It comes as a single balanced 3-class dataset, from which we discard the neutral reviews. We use an 80/10/10 split and clean whitespace characters in the same way as for the HARD dataset.

The TR movie and product reviews datasets by Demirtas and Pechenizkiy (2013) are already balanced and binary, so we only distribute the positive and negative instances evenly into training, development, and testing sets according to an 80/20/20 split.

Question Answering

For EN, we use the Stanford Question Answering Dataset (SQuAD) Version 1.1 (Rajpurkar et al., 2016). Each example in SQuAD consists of a context passage, a question, and one or more pairs of correct answer texts, and their starting positions within the context. All of the QA datasets we use follow exactly this format. SQuAD was released as a reading comprehension benchmark, so the test dataset is not publicly available.

For KO, we use the KorQuAD 1.0 dataset (Lim et al., 2019), which was crowdsourced from Korean Wikipedia article and introduced as a public benchmark in the same way as SQuAD. Therefore, only training and development splits of the dataset are publicly available.

For RU, we use the Sberbank Question Answering Dataset (SberQuAD), which was originally created for a competition by the Russian financial institution Sberbank and formally introduced to the scientific community by Efimov et al. (2020). Since the original splits for evaluation were never made public, the DeepPavlov team split the original training set into training and development splits[35]. We use these splits by DeepPavlov in our experiments.

We use the TQuAD[36] dataset for TR, which contains data on Turkish & Islamic science history and was released as part of the Teknofest 2018 Artificial Intelligence competition. It is split into training and development data.

For ZH, we use the Delta Reading Comprehension Dataset (DRCD; Shao et al., 2019), which was crowdsourced from Chinese Wikipedia articles. It is readily split into training, development, and test data.

For AR, FI, and ID, we extract all examples in their respective language from the multilingual TyDi QA secondary Gold Passage (GoldP) task dataset (Clark et al., 2020), which comes split into training and development data. Test sets are unavailable.

Considering that most of the datasets are only split into training and development splits, we decide not to use any test data in our QA experiments.

[34] https://www.kaggle.com/ilhamfp31/dataset-prosa/

[35] http://docs.deeppavlov.ai/en/master/features/models/squad.html

[36] https://tquad.github.io/turkish-nlp-qa-dataset/

Dependency Parsing & Part-of-Speech Tagging

We do not specifically preprocess any of the UD (Nivre et al., 2016, 2020; Zeman et al., 2020) treebanks. We give an overview of the treebanks we use in Table 8. We extract the head and dependency relation annotations for dependency parsing and the UPOS annotations for POS tagging. We do not perform additional preprocessing such as cleaning or filtering of the data.

A.5 Training Procedure of New Models

We pretrain our models on single Nvidia Tesla V100, A100, and Titan RTX GPUs with 32GB, 40GB, and 24GB of video memory, respectively. To support larger batch sizes, we train in mixed-precision (fp16) mode. Following Wu and Dredze (2020), we only use masked language modeling as pretraining objective and omit the next sentence prediction task as Liu et al. (2019) find it does not yield performance gains. We otherwise mostly follow the default pretraining procedure by Devlin et al. (2019).

We pretrain the new monolingual models (`wiki-mono`) from scratch for 1M steps with batch size 64. We choose a sequence length of 128 for the first 900,000 steps and 512 for the remaining 100,000 steps. In both phases, we warm up the learning rate to $1e-4$ over the first 10,000 steps, then decay linearly. We use the Adam optimizer with weight decay (AdamW) (Loshchilov and Hutter, 2019) with default hyper-parameters and a weight decay of 0.01. We enable whole word masking (Devlin et al., 2019) for the FI monolingual models, following the pretraining procedure for FinBERT (Virtanen et al., 2019). To lower computational requirements for the monolingual models with mBERT tokenizers, we remove all tokens from mBERT's vocabulary that do not appear in the pretraining data. We, thereby, obtain vocabularies of size

- 78,193 for AR
- 60,827 for FI
- 72,787 for ID
- 66,268 for KO
- 71,007 for TR,

which for all languages reduces the number of parameters in the embedding layer significantly, compared to the 119,547 word piece vocabulary of mBERT.

For the retrained mBERT models (`wiki-mbert-retrained`), we run masked language modeling for 250,000 steps (similar to Artetxe et al. (2020)) with batch size 64 and sequence length 512, otherwise using the same hyper-parameters as for the monolingual models. In order to retrain the embedding layer, we first resize it to match the vocabulary of the respective monolingual tokenizer. We initialize the positional embeddings, segment embeddings, and embeddings of special tokens (`[CLS]`, `[SEP]`, `[PAD]`, `[UNK]`, `[MASK]`) from mBERT, and reinitialize the remaining embeddings randomly. We freeze all parameters outside the embedding layer. For all pretraining runs, we set the random seed to 42.

B Further Analyses and Discussions

B.1 Tokenization Analysis

In our tokenization analysis in §4.2 of the main text, we only include the fertility and the proportion of continued words as they are sufficient to illustrate and quantify the differences between tokenizers. In support of the findings in §4.2 and for completeness, we provide additional tokenization statistics here.

For each tokenizer, Table 9 lists the respective vocabulary size and the proportion of its vocabulary also contained in mBERT. It shows that the tokenizers scoring lower in fertility (and accordingly performing better) than mBERT are often not adequately covered by mBERT's vocabulary. For instance, only 5.6% of the AraBERT (AR) vocabulary is covered by mBERT.

Figure 5 compares the proportion of unknown tokens (`[UNK]`) in the tokenized data. It shows that the proportion is generally extremely low, i.e., the tokenizers can typically split unknown words into known subwords.

Similar to the work by Ács (2019), the Figures 6 and 7 compare the tokenizations produced by the monolingual models and mBERT with the reference tokenizations provided by the human dataset annotators with respect to their sentence lengths. We find that the tokenizers scoring low in fertility and the proportion of continued words typically exhibit sentence length distributions much closer to the reference tokenizations by human UD annotators, indicating they are more capable than the mBERT tokenizer. Likewise, the monolingual models' and mBERT's sentence length distributions are

closer for languages with similar fertility and proportion of continued words, such as EN, JA, and ZH.

B.2 Correlation Analysis

To uncover some of the hidden patterns in our results (Tables 2 and 3), we perform a statistical analysis assessing the correlation between the individual factors (pretraining data size, subword fertility, proportion of continued words) and the downstream performance.

Figure 8 shows that both decreases in the proportion of continued words and the fertility correlate with an increase in downstream performance relative to fully fine-tuned mBERT across all tasks. The correlation is stronger for UDP and QA, where we found models with monolingual tokenizers to outperform their counterparts with the mBERT tokenizer consistently. The correlation is weaker for NER and POS tagging, which is also expected, considering the inconsistency of the results.

Somewhat surprisingly, the tokenizer metrics seem to be more indicative of high downstream performance than the size of the pretraining corpus. We believe that this in parts due to the overall poor performance of the uncased IndoBERT model, which we (in this case unfairly) compare to our cased `id-wiki-mono-mono` model. Therefore, we plot the same correlation matrix excluding ID in Figure 4.

Compared to Figure 8, the overall correlations for the proportion of continued words and the fertility remain mostly unaffected. In contrast, the correlation for the pretraining corpus size becomes much stronger, confirming that the subpar performance of IndoBERT is, in fact, an outlier in this scenario. Leaving out Indonesian also strengthens the indication that the performance in POS tagging correlates more with the data size than with the tokenizer, although we argue that this indication may be misleading. The performance gap is generally very minor in POS tagging. Therefore, the Spearman correlation coefficient, which only takes the rank into account, but not the absolute score differences, is particularly sensitive to changes in POS tagging performance.

Finally, we plot the correlation between the three metrics and the downstream performance under consideration of all languages and models, including the adapter-based fine-tuning settings, to gain an understanding of how pronounced their effects are in a more "noisy" setting.

As Figure 9 shows, the three factors still correlate with the downstream performance in a similar manner even when not isolated. This correlation tells us that even when there may be other factors that could have an influence, these three factors are still highly indicative of the downstream performance.

We also see that the correlation coefficients for the proportion of continued words and the fertility are nearly identical, which is expected based on the visual similarity of the respective plots (seen in Figures 2 and 3).

B.3 Effectiveness of Adapter-Based Fine-Tuning

We primarily included the adapter-based settings in our experimental framework based on the hypothesis that their effectiveness in cross-lingual settings (Pfeiffer et al., 2020c) could also transfer to monolingual settings.

However, our results suggest that adapters cannot close the performance gap between monolingual models and mBERT in most cases where it exists, despite their effectiveness in cross-lingual settings. The adapter-based variants of mBERT typically perform competitively with, and on rare occasions, even surpass the fully fine-tuned mBERT or monolingual models in performance. However, these scenarios seem to occur only when the performance gap is already small, in which case some randomness in the optimization process can already change the outcome qualitatively. Therefore, we argue that these individual occasions are not significant. Nevertheless, the competitive performance of adapters, for many languages and tasks even with the monolingual models trained on drastically more data, is a testament to the effectiveness of (parameter-)efficient deep learning approaches in NLP. Particularly in scenarios where maximum performance is not required and when computational resources are scarce, we highly suggest using mBERT with such adapter-based fine-tuning approaches.

Furthermore, we show that using language adapters (and invertible adapters) in conjunction with task adapters, as proposed by Pfeiffer et al. (2020c), is overall slightly more effective than using task adapters only. However, the effectiveness of language adapters seems also to be task-related. On the one hand, the dependency parsing results

favor the setting that includes a language adapter in addition to the task adapter. On the other hand, we observe the opposite for five out of nine languages in QA. In many cases, both settings perform equally well, or the performance gap is negligibly narrow.

Based on these results, it is difficult to recommend one setting over the other. Nevertheless, we suggest the following: For languages where language adapters are readily available on the AdapterHub (Pfeiffer et al., 2020b), there is generally little harm in using them. If language adapters are not available, it is most likely sufficient for monolingual tasks just to train new task adapters instead.

C Full Results

For compactness, we have only reported the performances of our models on the respective test datasets in the main text.[37] For completeness, we also include the full tables, including development (dev) dataset performances averaged over three random initializations, as described in §3. Table 10 shows the full results corresponding to Table 2 (initial results) and Table 11 shows the full results corresponding to Table 3 (results for our new models).

[37] Except for QA, where we do not use any test data

Language	Dataset	# Examples (Train / Dev)	# Words Total
AR	UD_Arabic-PADT	6075 / 909	254192
EN	UD_English-LinES	3176 / 1032	449977
	UD_English-EWT	12543 / 2002	
	UD_English-GUM	4287 / 784	
	UD_English-ParTUT	1781 / 156	
FI	UD_Finnish-FTB	14981 / 1875	324680
	UD_Finnish-TDT	12217 / 1364	
ID	UD_Indonesian-GSD	4477 / 559	110141
JA	UD_Japanese-GSD	7027 / 501	179571
KO	UD_Korean-GSD	4400 / 950	390369
	UD_Korean-Kaist	23010 / 2066	
RU	UD_Russian-GSD	3850 / 579	1130482
	UD_Russian-SynTagRus	48814 / 6584	
	UD_Russian-Taiga	3138 / 945	
TR	UD_Turkish-IMST	3664 / 988	47830
ZH	UD_Chinese-GSD	3997 / 500	222558
	UD_Chinese-GSDSimp	3997 / 500	

Table 4: Overview - UD v2.6 (Zeman et al., 2020) data used for our tokenizer analyses

Language	Dataset	Reference	Data Source	# Examples (Train/Dev/Test)	# Labels
AR	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	Wikipedia	20000 / 10000 / 10000	7
EN	CoNLL-2003	Tjong Kim Sang and De Meulder (2003)	News Articles	14041 / 3250 / 3453	8
FI	FiNER	Ruokolainen et al. (2020)	News Articles	13497 / 986 / 3512	6
ID	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	Wikipedia	20000 / 10000 / 10000	7
JA	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	Wikipedia	20202 / 10100 / 10113	7
KO	KMOU NER	[29]	News Articles	23056 / 468 / 463	22
RU	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	Wikipedia	20000 / 10000 / 10000	7
TR	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	Wikipedia	20000 / 10000 / 10000	7
ZH	Chinese Literature Dataset	Xu et al. (2017)	Literature	24270 / 1902 / 2844	7

Table 5: Named entity recognition dataset overview

Language	Dataset	Reference	Domain	# Examples (Train / Dev / Test)	# Labels	Balanced
AR	HARD	Elnagar et al. (2018)	Hotel Reviews	84558 / 10570 / 10570	2	Yes
EN	IMDb Movie Reviews	Maas et al. (2011)	Movie Reviews	20000 / 5000 / 25000	2	Yes
FI	—	—	—	—	—	—
ID	Indonesian Prosa	Purwarianti and Crisdayanti (2019)	Prose	6853 / 763 / 409	2	Yes
JA	Yahoo Movie Reviews	[31]	Movie Reviews	30545 / 3818 / 3819	2	Yes
KO	NSMC	[32]	Movie Reviews	120000 / 30000 / 50000	2	Yes
RU	RuReviews	Smetanin and Komarov (2019)	Product Reviews	48000 / 6000 / 6000	2	Yes
TR	Movie & Product Reviews	Demirtas and Pechenizkiy (2013)	Movie & Product Reviews	13009 / 1627 / 1629	2	Yes
ZH	ChnSentiCorp	[33]	Hotel Reviews	9600 / 1200 / 1200	2	Yes

Table 6: Sentiment analysis dataset overview

Language	Dataset	Reference	Domain	# Examples (Train / Dev)
AR	TyDiQA-GoldP	Clark et al. (2020)	Wiki	14805 / 921
EN	SQuAD v1.1	Rajpurkar et al. (2016)	Wiki	87599 / 10570
FI	TyDiQA-GoldP	Clark et al. (2020)	Wiki	6855 / 782
ID	TyDiQA-GoldP	Clark et al. (2020)	Wiki	5702 / 565
JA	—	—	—	—
KO	KorQuAD 1.0	Lim et al. (2019)	Wiki	60407 / 5774
RU	SberQuAD	Efimov et al. (2020)	Wiki	45328 / 5036
TR	TQuAD	[36]	—	8308 / 892
ZH	DRCD	Shao et al. (2019)	Wiki	26936 / 3524

Table 7: Question Answering dataset overview

Language	Dataset	# Examples (Train / Dev / Test)
AR	UD_Arabic-PADT	6075 / 909 / 680
EN	UD_English-EWT	12543 / 2002 / 2077
FI	UD_Finnish-FTB	14981 / 1875 / 1867
ID	UD_Indonesian-GSD	4477 / 559 / 557
JA	UD_Japanese-GSD	7027 / 501 / 543
KO	UD_Korean-GSD	4400 / 950 / 989
RU	UD_Russian-GSD	3850 / 579 / 601
TR	UD_Turkish-IMST	3664 / 988 / 983
ZH	UD_Chinese-GSD	3997 / 500 / 500

Table 8: Universal dependency parsing and part-of-speech tagging dataset (Zeman et al., 2020) overview

Lang	Tokenizer	Reference	Vocabulary Size	% Vocab in mBERT
MULTI	bert-base-multilingual-cased	Devlin et al. (2019)	119,547	100
AR	aubmindlab/bert-base-arabertv01	Antoun et al. (2020)	64,000	5.6
EN	bert-base-cased	Devlin et al. (2019)	28,996	66.4
FI	TurkuNLP/bert-base-finnish-cased-v1	Virtanen et al. (2019)	50,105	14.3
ID	indobenchmark/indobert-base-p2	Wilie et al. (2020)	30521	40.5
JA	cl-tohoku/bert-base-japanese-char	[17]	4,000	99.1
KO	KR-BERT-char-wordpiece	Lee et al. (2020)	16,424	47.4
RU	DeepPavlov/rubert-base-cased	Kuratov and Arkhipov (2019)	119,547	21.1
TR	dbmdz/bert-base-turkish-cased	Schweter (2020)	32,000	23.0
ZH	bert-base-chinese	Devlin et al. (2019)	21,128	79.4

Table 9: Comparison of vocabulary sizes of the selected monolingual BERT tokenizers and mBERT

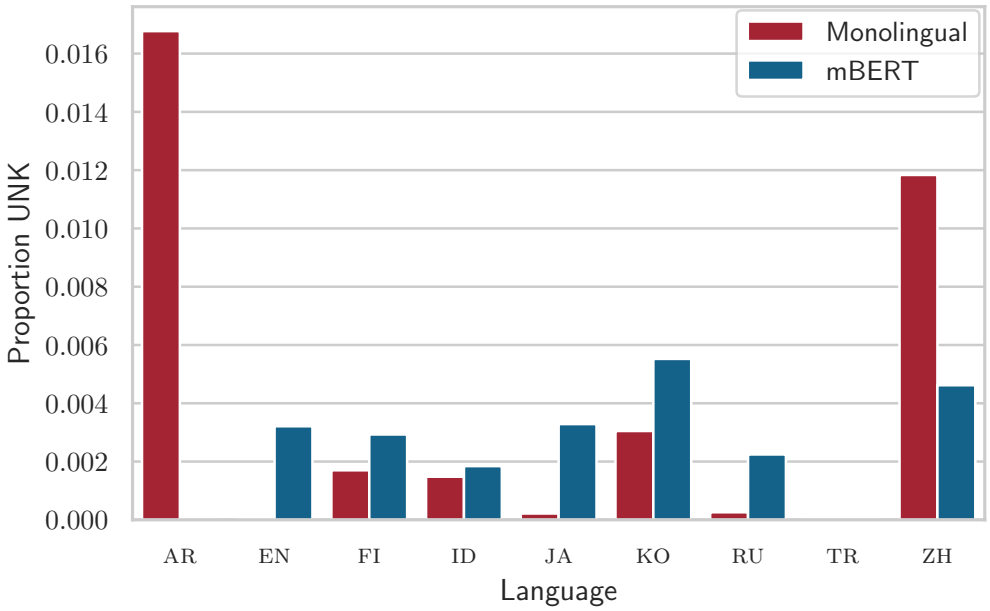
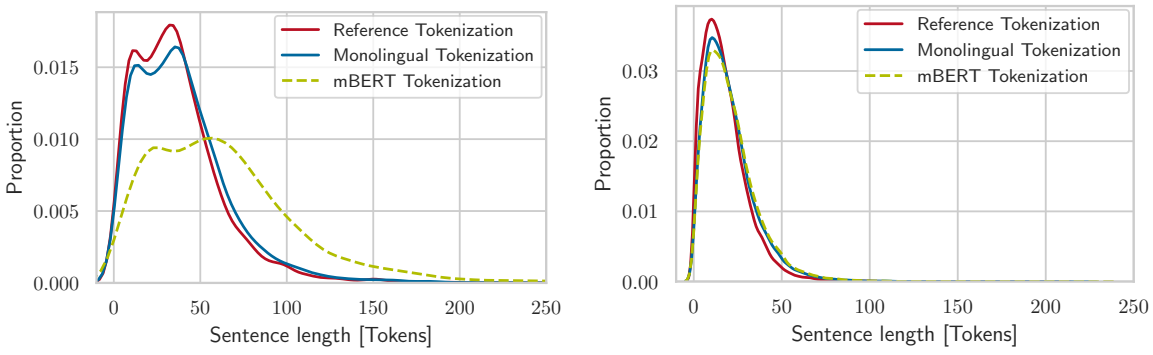
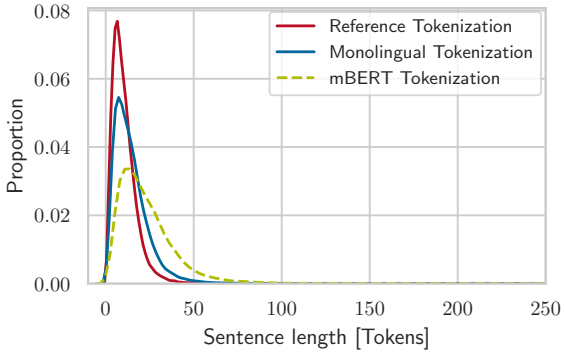


Figure 5: Proportion of unknown tokens in respective monolingual corpora tokenized by monolingual models vs. mBERT

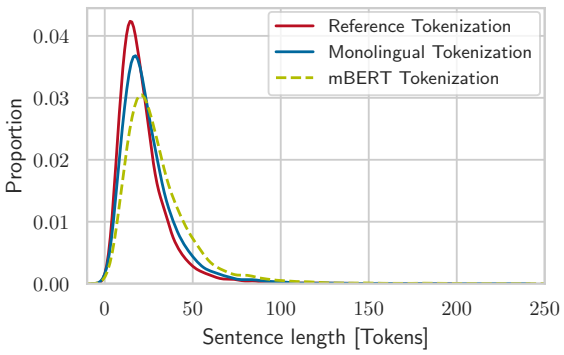


(a) Arabic – AR

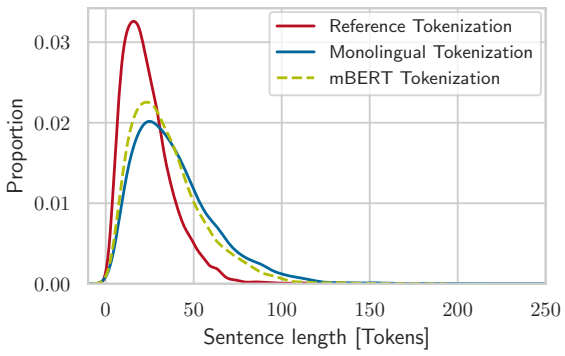
(b) English – EN



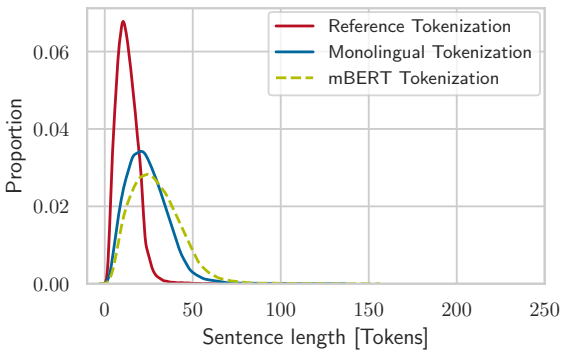
(c) Finnish – FI



(d) Indonesian – ID

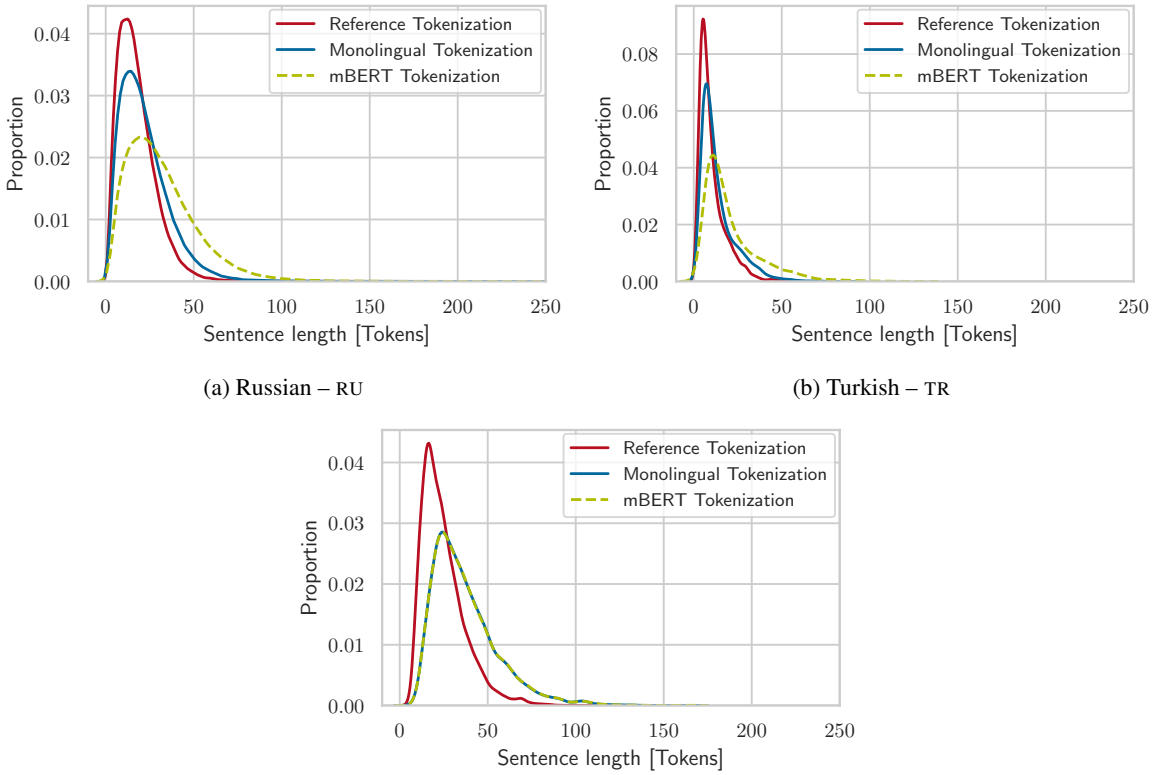


(e) Japanese – JA



(f) Korean – KO

Figure 6: Sentence length distributions of monolingual UD corpora tokenized by respective monolingual BERT models and mBERT, compared to the reference tokenizations by human UD treebank annotators - Part 1



(a) Russian – RU

(b) Turkish – TR

(c) Chinese – ZH

Figure 7: Sentence length distributions of monolingual UD corpora tokenized by respective monolingual BERT models and mBERT, compared to the reference tokenizations by human UD treebank annotators - Part 2

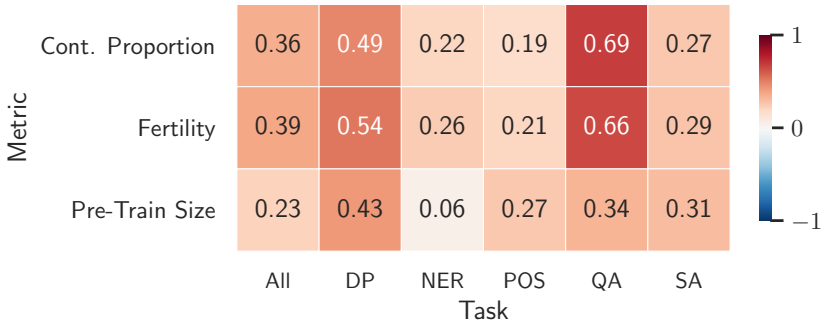


Figure 8: Spearman's ρ correlation of a relative decrease in the proportion of continued words (Cont. Proportion), a relative decrease in fertility, and a relative increase in pretraining corpus size with a relative increase in downstream performance over fully fine-tuned mBERT. For the proportion of continued words and the fertility, we consider fully fine-tuned mBERT, the `wiki-mono` models, and the `wiki-mbert-retrained` models. For the pretraining corpus size, we consider the original monolingual models and the `wiki-mono-mono` models.

Figure 9: Spearman's ρ correlation of a relative decrease in the proportion of continued words (Cont. Proportion), a relative decrease in fertility, and a relative increase in pretraining corpus size with a relative increase in downstream performance over fully fine-tuned mBERT. We consider all languages and models.

Lang	Model	NER		SA		QA	UDP		POS	
		Dev F_1	Test F_1	Dev Acc	Test Acc	Dev EM / F_1	Dev UAS / LAS	Test UAS / LAS	Dev Acc	Test Acc
AR	Monolingual	**91.5**	**91.1**	**96.1**	**95.9**	**68.3 / 82.4**	**89.4 / 85.0**	**90.1 / 85.6**	**97.5**	96.8
	mBERT	90.3	90.0	95.8	95.4	66.1 / 80.6	87.8 / 83.0	88.8 / 83.8	97.2	96.8
	+ $A^{\text{Lang, Task}}$	90.2	89.7	96.1	95.7	66.9 / 81.0	87.0 / 81.9	88.0 / 82.8	97.3	**96.8**
	+ A^{Task}	90.0	89.6	96.1	95.6	66.7 / 81.1	86.7 / 81.6	87.8 / 82.6	97.3	96.8
EN	Monolingual	95.4	**91.5**	**91.6**	**91.6**	80.5 / 88.0	**92.6 / 90.3**	**92.1 / 89.7**	**97.1**	**97.0**
	mBERT	**95.7**	91.2	90.1	89.8	**80.9 / 88.4**	92.1 / 89.6	91.6 / 89.1	97.0	96.9
	+ $A^{\text{Lang, Task}}$	95.5	91.4	89.9	89.4	80.1 / 87.7	91.6 / 88.9	91.3 / 88.7	96.9	96.7
	+ A^{Task}	95.6	90.5	90.1	89.8	79.9 / 87.6	91.6 / 88.9	91.0 / 88.3	96.8	96.7
FI	Monolingual	**93.3**	**92.0**	—	—	**69.9 / 81.6**	**95.7 / 93.9**	**95.9 / 94.4**	**98.1**	**98.4**
	mBERT	90.9	88.2	—	—	66.6 / 77.6	91.1 / 88.0	91.9 / 88.7	96.0	96.2
	+ $A^{\text{Lang, Task}}$	91.6	88.4	—	—	65.7 / 77.1	91.1 / 87.7	91.8 / 88.5	96.3	96.6
	+ A^{Task}	91.2	88.5	—	—	65.2 / 77.3	90.2 / 86.3	90.8 / 87.0	95.8	95.7
ID	Monolingual	90.9	91.0	**94.6**	**96.0**	66.8 / 78.1	84.5 / 77.4	85.3 / 78.1	92.0	92.1
	mBERT	**93.7**	**93.5**	93.1	91.4	**71.2** / 82.1	**85.0 / 78.4**	**85.9 / 79.3**	93.3	**93.5**
	+ $A^{\text{Lang, Task}}$	93.6	93.5	93.1	93.6	70.8 / 82.2	84.3 / 77.4	85.4 / 78.1	**93.6**	93.4
	+ A^{Task}	93.3	93.5	92.9	90.6	70.6 / **82.5**	83.7 / 76.5	84.8 / 77.4	93.5	93.4
JA	Monolingual	72.1	72.4	88.7	88.0	— / —	**96.0 / 94.7**	**94.7 / 93.0**	**98.3**	**98.1**
	mBERT	**73.4**	**73.4**	88.8	87.8	— / —	95.5 / 94.2	94.0 / 92.3	98.1	97.8
	+ $A^{\text{Lang, Task}}$	70.6	70.9	**89.4**	88.4	— / —	95.1 / 93.6	93.5 / 91.6	98.1	97.8
	+ A^{Task}	71.4	71.5	89.2	**88.6**	— / —	95.2 / 93.7	93.6 / 91.6	98.1	97.7
KO	Monolingual	**88.6**	**88.8**	**89.8**	**89.7**	**74.2 / 91.1**	**88.5 / 85.0**	**90.3 / 87.2**	**96.4**	**97.0**
	mBERT	87.3	86.6	86.7	86.7	69.7 / 89.5	86.9 / 83.2	89.2 / 85.7	95.8	96.0
	+ $A^{\text{Lang, Task}}$	87.3	86.2	86.6	86.3	70.0 / 89.8	85.9 / 81.6	88.3 / 84.3	96.0	96.2
	+ A^{Task}	87.1	86.2	86.7	86.5	69.8 / 89.7	85.5 / 81.1	87.8 / 83.9	95.9	96.2
RU	Monolingual	**91.9**	**91.0**	95.2	**95.2**	**64.3 / 83.7**	**92.4 / 90.1**	**93.1 / 89.9**	**98.6**	**98.4**
	mBERT	90.2	90.0	95.2	95.0	63.3 / 82.6	91.5 / 88.8	91.9 / 88.5	98.4	98.2
	+ $A^{\text{Lang, Task}}$	90.1	89.0	**95.2**	94.7	62.8 / 82.4	91.2 / 88.3	91.8 / 88.1	98.6	98.2
	+ A^{Task}	90.0	89.6	95.2	94.7	62.9 / 82.5	90.9 / 88.0	92.0 / 88.3	98.5	98.2
TR	Monolingual	93.1	92.8	**89.3**	**88.8**	**60.6 / 78.1**	**78.0 / 70.9**	**79.8 / 73.2**	**97.0**	**96.9**
	mBERT	**93.7**	**93.8**	86.4	86.4	57.9 / 76.4	72.6 / 65.2	74.5 / 67.4	95.5	95.7
	+ $A^{\text{Lang, Task}}$	93.3	93.5	86.2	84.8	56.9 / 75.8	71.1 / 63.0	73.0 / 64.7	96.0	95.9
	+ A^{Task}	93.0	93.0	86.1	83.9	55.3 / 75.1	70.4 / 62.0	72.4 / 64.1	95.5	95.7
ZH	Monolingual	**77.0**	**76.5**	**94.8**	**95.3**	82.3 / 89.3	**88.1 / 84.9**	**88.6 / 85.6**	**96.6**	**97.2**
	mBERT	76.0	76.1	93.1	93.8	82.0 / 89.3	87.1 / 83.7	88.1 / 85.0	96.1	96.7
	+ $A^{\text{Lang, Task}}$	75.6	75.4	94.0	94.8	82.1 / 89.4	86.0 / 82.1	87.3 / 83.8	96.1	96.4
	+ A^{Task}	75.4	75.2	93.8	94.1	**82.4 / 89.6**	85.8 / 81.9	87.5 / 83.9	96.1	96.5

Table 10: Full Results - Model Performances on Named Entity Recognition (NER), Sentiment Analysis (SA), Question Answering (QA), Universal Dependency Parsing (UDP, and Part-of-Speech Tagging (POS). Finnish (FI) SA and Japanese (JA) QA lack respective datasets.

Lang	Model	Tokenizer	NER		SA		QA	UDP		POS	
			Dev F_1	Test F_1	Dev Acc	Test Acc	Dev EM / F_1	Dev UAS / LAS	Test UAS / LAS	Dev Acc	Test Acc
AR	wiki-mono	mono	88.6	**91.7**	**96.0**	**95.6**	**67.7 / 81.6**	**88.4 / 83.7**	**89.2 / 84.4**	97.3	96.6
	wiki-mono	mBERT	**90.1**	90.0	95.9	95.5	64.1 / 79.4	87.8 / 83.2	88.8 / 84.0	**97.4**	**97.0**
	mBERT	mono	**91.9**	**91.2**	**95.9**	**95.4**	**66.9 / 81.8**	**88.2 / 83.5**	**89.3 / 84.5**	97.2	96.4
	mBERT	mBERT	90.3	90.0	95.8	95.4	66.1 / 80.6	87.8 / 83.0	88.8 / 83.8	**97.2**	**96.8**
FI	wiki-mono	mono	**91.9**	89.1	—	—	**66.9 / 79.5**	**93.6 / 91.0**	**93.7 / 91.5**	**97.0**	**97.3**
	wiki-mono	mBERT	91.8	**90.0**	—	—	65.1 / 77.0	93.1 / 90.6	93.6 / 91.5	96.2	97.0
	mBERT	mono	**91.0**	88.1	—	—	66.4 / **78.3**	**92.2 / 89.3**	**92.4 / 89.6**	**96.3**	**96.6**
	mBERT	mBERT	90.0	**88.2**	—	—	**66.6** / 77.6	91.1 / 88.0	91.9 / 88.7	96.0	96.2
ID	wiki-mono	mono	93.0	92.5	**93.9**	**96.0**	**73.1 / 83.6**	83.4 / 76.8	**85.0** / 78.5	**93.6**	**93.9**
	wiki-mono	mBERT	**93.3**	**93.2**	93.9	94.8	67.0 / 79.2	**84.0 / 77.4**	84.9 / **78.6**	93.4	93.6
	mBERT	mono	**93.8**	**93.9**	**94.4**	**94.6**	**74.1 / 83.8**	**85.5 / 78.8**	**86.4 / 80.2**	**93.5**	**93.8**
	mBERT	mBERT	93.7	93.5	93.1	91.4	71.2 / 82.1	85.0 / 78.4	85.9 / 79.3	93.3	93.5
KO	wiki-mono	mono	**87.9**	**87.1**	**89.0**	**88.8**	**72.8 / 90.3**	**87.9 / 84.2**	**89.8 / 86.6**	**96.4**	**96.7**
	wiki-mono	mBERT	86.9	85.8	87.3	87.2	68.9 / 88.7	86.9 / 83.2	88.9 / 85.6	**96.1**	96.4
	mBERT	mono	**87.9**	86.6	**88.2**	**88.1**	**72.9 / 90.2**	**87.9 / 83.9**	**90.1 / 87.0**	**96.2**	**96.5**
	mBERT	mBERT	87.3	**86.6**	86.7	86.7	69.7 / 89.5	86.9 / 83.1	89.2 / 85.7	95.8	96.0
TR	wiki-mono	mono	**93.5**	**93.4**	**87.5**	**87.0**	**56.2 / 73.7**	**74.4 / 67.3**	**76.1 / 68.9**	95.9	96.3
	wiki-mono	mBERT	93.2	93.3	85.8	84.8	55.3 / 72.5	73.2 / 66.0	75.3 / 68.3	**96.4**	**96.5**
	mBERT	mono	93.5	93.7	86.1	85.3	**59.4 / 76.7**	**74.7 / 67.6**	**77.1 / 70.2**	**96.1**	**96.3**
	mBERT	mBERT	**93.7**	**93.8**	**86.4**	**86.4**	57.9 / 76.4	72.6 / 65.2	74.5 / 67.4	95.5	95.7

Table 11: Full Results - Performances of our new `wiki-mono` and `wiki-mbert-retrained` models fine-tuned for Named Entity Recognition (NER), Sentiment Analysis (SA), Question Answering (QA), Universal Dependency Parsing (UDP), and Part-of-Speech Tagging (POS). We add the original fully fine-tuned mBERT and group counterparts w.r.t. tokenizer choice to facilitate a direct comparison between respective counterparts. mBERT model with mono tokenizer refers to `wiki-mbert-retrained` and mBERT model with mBERT tokenizer refers to the original fully fine-tuned mBERT.