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 and comprehensive 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 aim to establish, via fair and controlled comparisons, if a gap between the multilingual and the corresponding monolingual representation model of that language exists, and subsequently investigate the reason for any performance difference. To disentangle conflating factors, we train new monolingual models on the same data, with monolingually and multilingually trained tokenizers. We find that while the pretraining data size is an important factor, a designated monolingual tokenizer plays an equally important role in the downstream performance. Our results show that languages that 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 on large English corpora (e.g., BERT, RoBERTa, T5; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020), 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 in all NLP tasks. Concurrently, massively multilingual models with the same architectures and training procedures, covering more than 100 languages, have been proposed (e.g., mBERT, XLM-R, mT5; Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021).

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. 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 in 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 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 size. We further argue that

---

*Both authors contributed equally to this work.

†PR is now affiliated with the University of Copenhagen.

Our code is available at https://github.com/Adapter-Hub/hgiyt.

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 and validated that mBERT is under-trained (Rönnqvist et al., 2019; Wu and Dredze, 2020), providing evidence of improved performance when training monolingual models on more data, it is unclear if this is the only factor relevant for the performance of monolingual models. Another so far under-studied factor is the limited vocabulary size of multilingual models compared to the sum of tokens of all corresponding monolingual models. Our analyses investigating dedicated (i.e., language-specific) tokenizers reveal the importance of high-quality tokenizers for the performance of both model variants. We also shed light on the interplay of tokenization with other factors such as pretraining data size.

Contributions. **1)** We systematically compare monolingual with multilingual pretrained language models for 9 typologically diverse languages on 5 structurally different tasks. **2)** We train new monolingual models on equally sized datasets with different tokenizers (i.e., shared multilingual versus dedicated language-specific tokenizers) to disentangle the impact of pretraining data size from the vocabulary of the tokenizer. **3)** We isolate factors that contribute to a 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 factors on task performance. **4)** Our results suggest that monolingually adapted tokenizers can robustly improve monolingual performance of multilingual models.

2 Background and Related Work

Multilingual LMs. The widespread 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 by concatenating the 104 largest Wikipedias. Exponential smoothing was used when creating the subword vocabulary based on Word-Pieces (Wu et al., 2016) and a pretraining corpus. By oversampling underrepresented languages and undersampling overrepresented ones, it aims to counteract the imbalance of pretraining data sizes.

The final shared mBERT vocabulary comprises a total of 119,547 subword tokens.

Other multilingual models followed mBERT, such as XLM-R (Conneau et al., 2020). Concurrently, many studies analyzed 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 direct cross-lingual supervision (e.g., parallel or comparable data, translation dictionaries; Pires et al., 2019; Wu and Dredze, 2019; Artetxe et al., 2020; Hu et al., 2020; K et al., 2020).

However, recent work has also pointed to some fundamental limitations of multilingual LMs. Conneau et al. (2020) observe that, for a fixed model capacity, adding new languages increases cross-lingual performance up to a certain point, after which adding more languages results in performance drops. This phenomenon, termed the *curse of multilinguality*, can be attenuated by increasing the model capacity (Artetxe et al., 2020; Pfeiffer et al., 2020b; Chau et al., 2020) or through additional training for particular language pairs (Pfeiffer et al., 2020b; 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; Hu et al., 2020; Lauscher et al., 2020). Those languages remain underrepresented in the subword vocabulary and the model's shared representation space despite oversampling. Despite recent efforts to mitigate this issue (e.g., Chung et al. (2020) propose to cluster and merge the vocabularies of similar languages, before defining a joint vocabulary across all languages), the multilingual LMs still struggle with balancing their parameters across many languages.

Monolingual versus Multilingual LMs. New monolingual language-specific models also emerged for many languages, following BERT's architecture and pretraining procedure. 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), 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 LMs compare across languages.

Nozza et al. (2020) extracted task results from the respective papers on monolingual BERTs to facilitate an overview of monolingual models and their comparison to mBERT.[1] However, they have not verified the scores, nor have they performed a controlled impartial comparison.

Vulić et al. (2020) probed mBERT and monolingual BERT models across six typologically diverse languages for lexical semantics. They show that pretrained monolingual BERT models encode significantly more lexical information than mBERT.

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

Mulcaire et al. (2019) compared monolingual and bilingual ELMo (Peters et al., 2018) LMs across three downstream tasks, finding that contextualized representations from the bilingual models can improve monolingual task performance relative to their monolingual counterparts.[2] However, it is unclear how their findings extend to *massively* multilingual LMs potentially suffering from the curse of multilinguality.

Rönnqvist et al. (2019) 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. They speculate that mBERT is under-trained with respect to individual languages. However, their set of tasks is limited, and their language sample is typologically narrow; 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.

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 tokenizers and pretraining data, we can conduct unbiased and fair comparisons.

3.1 Language and Task Selection

Our 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 language-specific biases (Gerz et al., 2018; Ponti et al., 2019; Joshi et al., 2020). Following recent work in multilingual NLP that 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. We achieve task diversity and generalizability by selecting a combination of tasks driven by lower-level syntactic and higher-level semantic features (Lauscher et al., 2020).

Finally, we select a set of 9 languages from 8 language families, as listed in Table 1.[3] We evaluate mBERT and monolingual 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).[4]

---

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

[2]Mulcaire et al. (2019) clearly differentiate between *multilingual* and *polyglot* models. Their definition of polyglot models is in line with what we term multilingual models.

[3]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. Additional information regarding our selection of pretrained models is available in Appendix A.1.

[4]Information on which datasets are associated with which language and the dataset sizes (examples per split) are provided in Appendix A.4.

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[5]
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.

Named Entity Recognition (NER). We rely on: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), FiNER (Ruokolainen et al., 2020), Chinese Literature (Xu et al., 2017), KMOU NER,[6] WikiAnn (Pan et al., 2017; Rahimi et al., 2019).

Sentiment Analysis (SA). We employ: HARD (Elnagar et al., 2018), IMDb Movie Reviews (Maas et al., 2011), Indonesian Prosa (Purwarianti and Crisdayanti, 2019), Yahoo Movie Reviews,[7] NSMC,[8] RuReviews (Smetanin and Komarov, 2019), Turkish Movie and Product Reviews (Demirtas and Pechenizkiy, 2013), ChnSentiCorp.[9]

Question Answering (QA). We use: SQuADv1.1 (Rajpurkar et al., 2016), KorQuAD 1.0 (Lim et al., 2019), SberQuAD (Efimov et al., 2020), TQuAD,[10] DRCD (Shao et al., 2019), TyDiQA-GoldP (Clark et al., 2020).

Dependency Parsing (UDP). We rely on Universal Dependencies (Nivre et al., 2016, 2020) v2.6 (Zeman et al., 2020) for all languages.

Part-of-Speech Tagging (POS). We again utilize Universal Dependencies v2.6.

3.2 Task-Based Fine-Tuning

Fine-Tuning Setup. For all tasks besides UDP, we use the standard fine-tuning setup of Devlin et al. (2019). For UDP, we use a transformer-based variant (Glavaš and Vulić, 2021) of the standard deep biaffine attention dependency parser (Dozat and Manning, 2017). We distinguish between fully fine-tuning a monolingual BERT model and fully fine-tuning mBERT on the task. For both settings, we average scores over three random initializations on the development set. On the test set, we report

[5] https://github.com/cl-tohoku/bert-japanese
[6] https://github.com/kmounlp/NER
[7] https://github.com/dennybritz/sentiment-analysis
[8] https://www.lucypark.kr/docs/2015-pyconkr/#39
[9] https://github.com/pengming617/bert_classification
[10] https://tquad.github.io/turkish-nlp-qa-dataset/

Lg	Model	NER Test F_1	SA Test Acc	QA Dev EM / F_1	UDP Test UAS / LAS	POS 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
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
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
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**
JA	Monolingual	72.4	**88.0**	—— / ——	**94.7 / 93.0**	**98.1**
	mBERT	**73.4**	87.8	—— / ——	94.0 / 92.3	97.8
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
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
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
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
AVG	Monolingual	**87.4**	**92.4**	**70.8 / 84.0**	**90.0 / 86.3**	**96.9**
	mBERT	87.0	91.0	69.7 / 83.3	88.4 / 84.4	96.4

Table 2: Performance 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.

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 AdamW (Kingma and Ba, 2015) in all experiments, with a learning rate of $3e-5$.[11] We train for 10 epochs with early stopping (Prechelt, 1998).[12]

---

[11] Preliminary experiments indicated this to be a well performing learning rate. 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 of $5e-4$ works best for adapter-based fine-tuning (see later) since the task adapter parameters are learned from scratch (i.e., they are randomly initialized).

[12] We evaluate a model every 500 gradient steps on the development set, saving the best-performing model based on the respective evaluation measures. We terminate training if no performance gains are observed within five consecutive evaluation runs (= 2,500 steps). For QA and UDP, we use the F_1 scores and LAS, respectively. For FI and ID QA, we train for 20 epochs due to slower convergence. 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.[13] We find that the performance gap between monolingual models and mBERT does exist to a large extent, confirming anecdotal evidence 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 except 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. In what follows, we seek to understand the causes of this behavior in relation to different factors such as tokenizers, corpora sizes, as well as languages and tasks in consideration.

4 Tokenizer versus 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, we estimate the latter by the total number of words across all articles listed for each Wiki.[14] For the monolingual LMs, we extract information on pretraining data from the model documentation. If no exact numbers are explicitly stated, and the pretraining corpora are unavailable, we make estimations based on the information provided by the authors.[15] The statistics are provided in Figure 1a. For EN, JA, RU, and ZH, both the respective monolingual BERT and mBERT were trained on similar amounts of monolingual data. On the other hand, monolingual BERTs of AR, ID, FI, KO, and TR were trained on 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 parameter budget that it can allocate for each of its 104 languages in its vocabulary. In addition, monolingual tokenizers are typically trained by native-speaking experts who are 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 v2.6 treebanks and compute two metrics (Ács, 2019).[16] First, the subword *fertility* measures the average number of subwords produced per tokenized word. A minimum fertility of 1 means that the tokenizer's vocabulary contains every single word in the text. We plot the fertility scores in Figure 1b. 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 are over-segmented. mBERT's fertility is the lowest for EN; this is due to mBERT having seen the most data in this language during training, as well as English being morphologically poor in contrast to languages such as AR, FI, RU, or TR.[17]

The second metric we employ is the proportion of words 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, this metric 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 1c show similar trends as with the fertility statistic. In addition to AR, FI, KO, RU, and TR, which already displayed differences in fertility, mBERT also produces a proportion of continued words more than twice as high as the monolingual model for ID.[18]

---

[13]See Appendix Table 8 for the results on development sets.
[14]Based on the numbers from
https://meta.m.wikimedia.org/wiki/List_of_Wikipedias
[15]We provide further details in Appendix A.2.

[16]We provide further details in Appendix A.3.
[17]The JA model is the only monolingual BERT with a fertility score higher than mBERT; its tokenizer is character-based and thus by design produces the maximum number of subwords.
[18]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.

(a) Pretraining corpus size   (b) Subword fertility   (c) Proportion of continued words
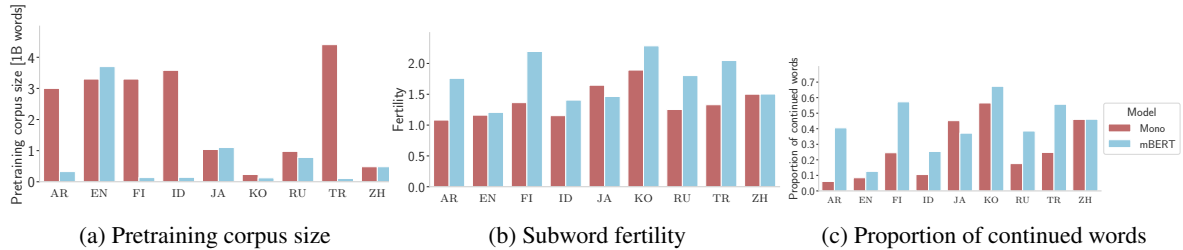
Figure 1: Comparison of monolingual models with mBERT w.r.t. pretraining corpus size (measured in billions of words), subword fertility (i.e., the average number of subword tokens produced per tokenized word (Ács, 2019)), and proportion of continued words (i.e., words split into multiple subword tokens (Ács, 2019)).

4.3 New Pretrained Models

The differences in pretraining corpora and tokenizer statistics 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 vs. EN, JA, ZH).[19] This implies that both the data size and the tokenizer are among the main driving forces of downstream task performance. 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 in tokenization and pretraining data size) on Wikipedia data.

We train four model variants for each language. First, we train two new monolingual BERT models on the same data, one with the original monolingual tokenizer (MONOMODEL-MONOTOK) and one with the mBERT tokenizer (MONOMODEL-MBERTTOK).[20] Second, similar to Artetxe et al. (2020), we retrain the embedding layer of mBERT, once with the respective monolingual tokenizer (MBERTMODEL-MONOTOK) and once with the mBERT tokenizer (MBERTMODEL-MBERTTOK). We freeze the transformer and only retrain the embedding weights, thus largely preserving mBERT's multilinguality. The reason we retrain mBERT's embedding layer with its own tokenizer is to further eliminate confounding factors when comparing to the version of mBERT with monolingually retrained embeddings. By comparing models

trained on the same amount of data, but with different tokenizers (MONOMODEL-MONOTOK vs. MONOMODEL-MBERTTOK, MBERTMODEL-MBERTTOK vs. MBERTMODEL-MONOTOK), we disentangle the effect of the dataset size from the tokenizer, both with monolingual and multilingual LM variants.

Pretraining Setup. We pretrain new BERT models for each language on its respective Wikipedia dump.[21] 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 to segment the extracted text passages into texts with document, sentence, and word boundaries.

Following Liu et al. (2019); Wu and Dredze (2020), we only use the masked language modeling (MLM) objective and omit the next sentence prediction task. Besides that, we largely follow the default pretraining procedure by Devlin et al. (2019). We pretrain the new monolingual LMs (MONOMODEL-*) from scratch for 1M steps.[22] 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 (MBERTMODEL-*), we train for 250,000 steps following Artetxe et al. (2020).[23] We freeze all parameters outside the embedding layer.[24]

Results. We perform the same evaluations on downstream tasks for our new models as described

---

[19]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 is because IndoBERT is uncased, whereas the remaining models are cased.

[20]The only exception is ID; 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).

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

[22]The batch size is 64; the sequence length is 128 for the first 900,000 steps, and 512 for the remaining 100,000 steps.

[23]We train with batch size 64 and sequence length 512, otherwise using the same hyper-parameters as for the monolingual models.

[24]For more details see Appendix A.5.

Lg	Model	NER Test F_1	SA Test Acc	QA Dev EM / F_1	UDP Test UAS / LAS	POS Test Acc
	Monolingual	91.1	**95.9**	**68.3 / 82.4**	90.1 / 85.6	96.8
	MONOMODEL-MONOTOK	**91.7**	95.6	67.7 / 81.6	89.2 / 84.4	96.6
	MONOMODEL-mBERTTOK	90.0	95.5	64.1 / 79.4	88.8 / 84.0	**97.0**
AR	mBERTMODEL-MONOTOK	91.2	95.4	66.9 / 81.8	89.3 / 84.5	96.4
	mBERTMODEL-mBERTTOK	89.7	95.6	66.3 / 80.7	89.1 / 84.2	96.8
	mBERT	90.0	95.4	66.1 / 80.6	88.8 / 83.8	96.8
	Monolingual	**92.0**	——	**69.9 / 81.6**	**95.9 / 94.4**	**98.4**
	MONOMODEL-MONOTOK	89.1	——	66.9 / 79.5	93.7 / 91.5	97.3
	MONOMODEL-mBERTTOK	90.0	——	65.1 / 77.0	93.6 / 91.5	97.0
FI	mBERTMODEL-MONOTOK	88.1	——	66.4 / 78.3	92.4 / 89.6	96.6
	mBERTMODEL-mBERTTOK	88.1	——	65.9 / 77.3	92.2 / 89.4	96.7
	mBERT	88.2	——	66.6 / 77.6	91.9 / 88.7	96.2
	Monolingual	91.0	96.0	66.8 / 78.1	85.3 / 78.1	92.1
	MONOMODEL-MONOTOK	92.5	**96.0**	73.1 / 83.6	85.0 / 78.5	**93.9**
	MONOMODEL-mBERTTOK	93.2	94.8	67.0 / 79.2	84.9 / 78.6	93.6
ID	mBERTMODEL-MONOTOK	**93.9**	94.6	**74.1 / 83.8**	**86.4 / 80.2**	93.8
	mBERTMODEL-mBERTTOK	**93.9**	94.6	71.9 / 82.7	86.2 / 79.6	93.7
	mBERT	93.5	91.4	71.2 / 82.1	85.9 / 79.3	93.5
	Monolingual	**88.8**	**89.7**	**74.2 / 91.1**	**90.3 / 87.2**	**97.0**
	MONOMODEL-MONOTOK	87.1	88.8	72.8 / 90.3	89.8 / 86.6	96.7
	MONOMODEL-mBERTTOK	85.8	87.2	68.9 / 88.7	88.9 / 85.6	96.4
KO	mBERTMODEL-MONOTOK	86.6	88.1	72.9 / 90.2	90.1 / 87.0	96.5
	mBERTMODEL-mBERTTOK	86.2	86.6	69.3 / 89.3	89.2 / 85.9	96.2
	mBERT	86.6	86.7	69.7 / 89.5	89.2 / 85.7	96.0
	Monolingual	92.8	**88.8**	**60.6 / 78.1**	**79.8 / 73.2**	**96.9**
	MONOMODEL-MONOTOK	93.4	87.0	56.2 / 73.7	76.1 / 68.9	96.3
	MONOMODEL-mBERTTOK	93.3	84.8	55.3 / 72.5	75.3 / 68.3	96.5
TR	mBERTMODEL-MONOTOK	93.7	85.3	59.4 / 76.7	77.1 / 70.2	96.3
	mBERTMODEL-mBERTTOK	**93.8**	86.1	58.7 / 76.6	76.2 / 69.2	96.3
	mBERT	93.8	86.4	57.9 / 76.4	74.5 / 67.4	95.7
	Monolingual	91.1	92.6	68.0 / 82.3	88.3 / 83.7	96.2
	MONOMODEL-MONOTOK	90.8	91.9	67.3 / 81.7	86.8 / 82.0	96.2
	MONOMODEL-mBERTTOK	90.5	90.6	64.1 / 79.4	86.3 / 81.6	96.1
AVG	mBERTMODEL-MONOTOK	90.7	90.9	68.0 / 82.2	87.1 / 82.3	95.9
	mBERTMODEL-mBERTTOK	90.3	90.7	66.4 / 81.3	86.6 / 81.7	95.9
	mBERT	90.4	90.0	66.3 / 81.2	86.1 / 81.0	95.6

Table 3: Performance of our new MONOMODEL-* and mBERTMODEL-* models (see §A.5) fine-tuned for the NER, SA, QA, UDP, and POS tasks (see §3.1), compared to the monolingual models from prior work and fully fine-tuned mBERT. We group model counterparts w.r.t. tokenizer choice to facilitate a direct comparison between respective counterparts. We use development sets only for QA. **Bold** denotes best score across all models for a given language and task. Underlined denotes best score compared to its respective counterpart.

in §3, and report the results in Table 3.[25]

The results indicate 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 performance gap is the smallest for POS tagging (at most 0.4% accuracy). We observe the

[25]Full results including development set scores are available in Table 9 of the Appendix.

largest gaps for QA (6.1 EM / 4.4 F_1 in ID), SA (2.2% accuracy in TR), and NER (1.7 F_1 in AR). Although the only language in which the monolingual counterpart always comes out on top is KO, the multilingual counterpart comes out on top at most 3/10 times (for AR and TR) in the other languages. The largest decrease in performance of a monolingual tokenizer relative to its multilingual counterpart is found for SA in TR (0.8% accuracy).

Overall, we find that for 38 out of 48 task, model, and language combinations, the monolingual tokenizer outperforms the mBERT counterpart. We were able to improve the monolingual performance of the original mBERT for 20 out of 24 languages and tasks by only replacing the tokenizer and, thus, leveraging a specialized monolingual version. Similar to how the chosen method of tokenization affects neural machine translation quality (Domingo et al., 2019), these results establish that, in fact, the designated pretrained tokenizer plays a fundamental role in the monolingual downstream task performance of contemporary LMs.

In 18/24 language and task settings, the monolingual model from prior work (trained on more data) outperforms its corresponding MONOMODEL-MONOTOK model. 4/6 settings in which our MONOMODEL-MONOTOK model performs better are found for ID, where IndoBERT uses an uncased tokenizer and our model uses a cased one, which may affect the comparison. Expectedly, these results strongly indicate that data size plays a major role in downstream performance and corroborate prior research findings (Liu et al., 2019; Conneau et al., 2020; Zhang et al., 2020, *inter alia*).

4.4 Adapter-Based Training

Another way to provide more language-specific capacity to a multilingual LM beyond a dedicated tokenizer, thereby potentially making gains in monolingual downstream performance, is to introduce adapters (Pfeiffer et al., 2020b,c; Üstün et al., 2020), a small number of additional parameters at every layer of a pretrained model. To train adapters, usually all pretrained weights are frozen, while only the adapter weights are fine-tuned.[26] The adapter-based approaches thus offer increased efficiency and modularity; it is crucial to verify to which extent our findings extend to the more efficient and

[26]Pfeiffer et al. (2020b) propose to stack task-specific adapters on top of language adapters and extend this approach in Pfeiffer et al. (2020c) by additionally training new embeddings for the target language.

Lg	Model	NER Test F_1	SA Test Acc	QA Dev EM / F_1	UDP Test UAS / LAS	POS Test Acc
AR	mBERT	90.0	95.4	66.1 / 80.6	**88.8 / 83.8**	**96.8**
	+ A^Task	89.6	95.6	66.7 / 81.1	87.8 / 82.6	**96.8**
	+ A^Task + A^Lang	89.7	**95.7**	66.9 / 81.0	88.0 / 82.8	**96.8**
	+ A^Task + A^Lang + MonoTok	**91.1**	**95.7**	**67.7 / 82.1**	88.5 / 83.4	96.5
FI	mBERT	88.2	——	66.6 / 77.6	91.9 / 88.7	96.2
	+ A^Task	**88.5**	——	65.2 / 77.3	90.8 / 87.0	95.7
	+ A^Task + A^Lang	88.4	——	65.7 / 77.1	91.8 / 88.5	96.6
	+ A^Task + A^Lang + MonoTok	88.1	——	**66.7 / 79.0**	**92.8 / 90.1**	**97.3**
ID	mBERT	**93.5**	91.4	71.2 / 82.1	**85.9 / 79.3**	**93.5**
	+ A^Task	**93.5**	90.6	70.6 / 82.5	84.8 / 77.4	93.4
	+ A^Task + A^Lang	**93.5**	93.6	70.8 / 82.2	85.4 / 78.1	93.4
	+ A^Task + A^Lang + MonoTok	93.4	**93.8**	**74.4 / 84.4**	85.1 / 78.3	**93.5**
KO	mBERT	**86.6**	86.7	69.7 / 89.5	**89.2 / 85.7**	96.0
	+ A^Task	86.2	86.5	69.8 / 89.7	87.8 / 83.9	96.2
	+ A^Task + A^Lang	86.2	86.3	70.0 / 89.8	88.3 / 84.3	96.2
	+ A^Task + A^Lang + MonoTok	86.5	**87.9**	**73.1 / 90.4**	88.9 / 85.2	**96.5**
TR	mBERT	**93.8**	**86.4**	57.9 / 76.4	74.5 / 67.4	95.7
	+ A^Task	93.0	83.9	55.3 / 75.1	72.4 / 64.1	95.7
	+ A^Task + A^Lang	93.5	84.8	56.9 / 75.8	73.0 / 64.7	95.9
	+ A^Task + A^Lang + MonoTok	92.7	85.3	**60.0 / 77.0**	**75.7 / 68.1**	**96.3**
AVG	mBERT	**90.4**	90.0	66.3 / 81.2	**86.0 / 81.0**	95.6
	+ A^Task	90.2	89.2	65.5 / 81.1	84.7 / 79.0	95.6
	+ A^Task + A^Lang	90.3	90.1	66.1 / 81.2	85.3 / 79.7	95.8
	+ A^Task + A^Lang + MonoTok	**90.4**	**90.7**	**68.4 / 82.6**	86.2 / 81.0	**96.0**

Table 4: Performance on the different tasks leveraging mBERT with different adapter components (see §4.4).

more versatile adapter-based fine-tuning setup.

We evaluate the impact of different adapter components on the downstream task performance and their complementarity with monolingual tokenizers in Table 4.[27] Here, $+A^{Task}$ and $+A^{Lang}$ implies adding task- and language-adapters respectively, whereas +MonoTok additionally includes a new embedding layer. As mentioned, we only fine-tune adapter weights on the downstream task, leveraging the adapter architecture proposed by Pfeiffer et al. (2021). For the $+A^{Task} + A^{Lang}$ setting we leverage pretrained language adapter weights available at AdapterHub.ml (Pfeiffer et al., 2020a). Language adapters are added to the model and frozen while only task adapters are trained on the target task. For the $+A^{Task} + A^{Lang} +$ MonoTok we train language adapters and new embeddings with the corresponding monolingual tokenizer equally as described in the previous section (e.g. MBERTMODEL-MONOTOK), task adapters are trained with a learning rate of $5e - 4$ and 30 epochs with early stopping.

Results. Similar to previous findings, adapters improve upon mBERT in 18/24 language, and task settings, 13 of which can be attributed to the improved MBERTMODEL-MONOTOK tokenizer. Figure 2 illustrates the average performance of the different adapter components in comparison to the monolingual models. We find that adapters with dedicated tokenizers reduce the performance gap con-

---
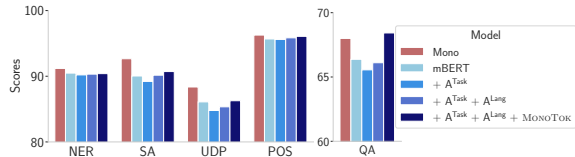
[27] See Appendix Table 10 for the results on dev sets.

Figure 2: Task performance averaged over all languages for different models: fully fine-tuned monolingual (**Mono**), fully fine-tuned mBERT (**mBERT**), mBERT with task adapter ($+\mathbf{A^{Task}}$), with task and language adapter ($+\mathbf{A^{Task}} + \mathbf{A^{Lang}}$), with task and language adapter and embedding layer retraining ($+\mathbf{A^{Task}} + \mathbf{A^{Lang}} +$ MONOTOK).

siderably without leveraging more training data, and even outperform the monolingual models in QA. This finding shows that adding additional language-specific capacity to existing multilingual LMs, which can be achieved with adapters in a portable and efficient way, is a viable alternative to monolingual pretraining.

5 Further Analysis

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, *inter alia*). However, the broad scope of our experiments reveals certain nuances that were previously undiscovered. Unlike prior work, which primarily attributes gaps in performance to mBERT being under-trained (Rönnqvist et al., 2019; Wu and Dredze, 2020), our disentangled results (Table 3) suggest that a large portion of existing performance gaps can be attributed to the capability of the tokenizer.

With monolingual tokenizers with lower fertility and proportion-of-continued-words values than the mBERT tokenizer (such as for AR, FI, ID, KO, TR), consistent gains can be achieved, irrespective of whether the LMs are monolingual (the MONOMODEL-* comparison) or multilingual (a comparison of MBERTMODEL-* variants).

Whenever the differences between monolingual models and mBERT with respect to the tokenizer properties and the pretraining corpus size are small (e.g., for EN, JA, and ZH), the performance gap is typically negligible. In QA, we even find mBERT to be favorable for these languages. Therefore, we conclude that monolingual models are not superior to multilingual ones per se, but gain advantage in direct comparisons by incorporating more pretraining data and using language-adapted tokenizers.
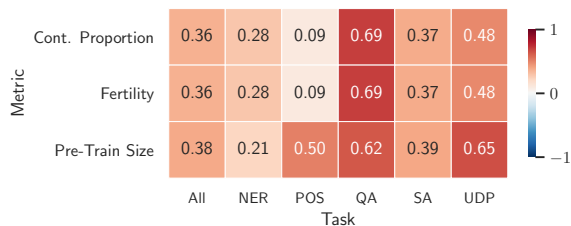
Figure 3: Spearman's ρ correlation of a relative decrease in the proportion of continued words (Cont. Proportion), a relative decrease in fertility, and a relative increase in pretraining corpus size with a relative increase in downstream performance over fully fine-tuned mBERT. For the proportion of continued words and the fertility, we consider fully fine-tuned mBERT, the MONOMODEL-* models, and the MBERTMODEL-* models. For the pretraining corpus size, we consider the original monolingual models and the MONOMODEL-MONOTOK models. We exclude the ID models (see Appendix B.2 for the clarification).

Correlation Analysis. To uncover additional patterns in our results (Tables 2, 3, 4), 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 relations not immediately evident by looking at the tables.

Figure 3 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 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.[28]

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 ability of representing a language plays a crucial role; Consequently, choosing a sub-optimal tokenizer typically results in deteriorated downstream performance.

---

[28]For further information, see Appendix B.2.

6 Conclusion

We have conducted the first comprehensive empirical investigation concerning the monolingual performance of monolingual and multilingual language models (LMs). While our results support the existence of a performance gap in most but not all languages and tasks, further analyses revealed that the gaps are often substantially smaller than what was previously assumed. The gaps exist in certain languages due to the discrepancies in **1)** pretraining data size, and **2)** chosen tokenizers, and the level of their adaptation to the target language.

Further, we have disentangled the impact of pretrained corpora size from the influence of the tokenizers on the downstream task performance. We have trained new monolingual LMs on the same data, but with two different tokenizers; one being the dedicated tokenizer of the monolingual LM provided by native speakers; the other being the automatically generated multilingual mBERT tokenizer. We have found that for (almost) every task and language, the use of monolingual tokenizers outperforms the mBERT tokenizer.

Consequently, in line with recent work by Chung et al. (2020), our results suggest that investing more effort into **1)** improving the balance of individual languages' representations in the vocabulary of multilingual LMs, and **2)** providing language-specific adaptations and extensions of multilingual tokenizers (Pfeiffer et al., 2020c) can reduce the gap between monolingual and multilingual LMs. Another promising future research direction is completely disposing of any (language-specific or multilingual) tokenizers during pretraining (Clark et al., 2021).

Our code, pretrained models, and adapters are available at https://github.com/Adapter-Hub/hgiyt.

Acknowledgments

References

Judit Ács. 2019. Exploring BERT's Vocabulary. *Blog Post*.

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.

Giusepppe Attardi. 2015. Wikiextractor. *GitHub Repository*.

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

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

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.

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

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.

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)*, pages 9:1–8, Chicago, 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.

Miguel Domingo, Mercedes García-Martínez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2019. How much does tokenization affect neural machine translation? *arXiv preprint*.

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. In *CLEF 2020: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 3–15. Springer, Cham, Switzerland.

Ashraf Elnagar, Yasmin S. Khalifa, and Anas Einea. 2018. Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications. In *Intelligent Natural Language Processing: Trends and Applications*, pages 35–52. Springer, 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. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.

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, 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. *arXiv preprint*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. 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. Association for Computational Linguistics.

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.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, Minneapolis, Minnesota. 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*.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages

2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. 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. 2020b. 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. 2020c. 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.

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*, pages 55–69. Springer, Berlin, Germany.

Ayu Purwarianti and Ida Ayu Putu Ari 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):1–67.

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.

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)*, pages 482–486, Moscow, Russia. IEEE.

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'16)*, 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*, pages 142–147, Edmonton, Canada. Association for Computational Linguistics.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 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. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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 Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, 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 Pate-

juk, 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.

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

A   Reproducibility

A.1   Pretrained Models

All of the pretrained language models we use are available on the HuggingFace model hub[29] and compatible with the HuggingFace transformers Python library (Wolf et al., 2020). Table 5 displays the model hub identifiers of our selected models.

A.2   Estimating the Pretraining Corpora Sizes

Since mBERT was pretrained on the entire Wikipedia dumps of all languages it covers (Devlin et al., 2019), we estimate the language-specific shares of the mBERT pretraining corpus by word counts of the respective raw Wikipedia dumps, according to numbers obtained from Wikimedia[30]: 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.[31] Devlin et al. (2019) only included text passages from the articles, and used older Wikipedia dumps, so these numbers should serve as upper limits, yet be reasonably accurate. For the monolingual models, we rely on information provided by the authors.[32]

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 6.

A.4   Fine-Tuning Datasets

We list the datasets we used, including the number of examples per dataset split, in the Table 7.

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 (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 (MONOMODEL-*) 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 (AR), 60,827 (FI), 72,787 (ID), 66,268 (KO), and 71,007 (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 (i.e., MBERTMODEL-*), we run MLM 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 tokenizer. For the MBERTMODEL-MBERTTOK models, we use the mBERT tokenizers with reduced vocabulary as outlined above. 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.

A.6   Code

Our code with usage instructions for fine-tuning, pretraining, data preprocessing, and calculating the tokenizer statistics is available at https://github.com/Adapter-Hub/hgiyt. The repository also contains further links to a collection of our new pretrained models and language adapters.

---

[29] https://huggingface.co/models

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

[31] We obtained the numbers for ID and TR on Dec 10, 2020 and for the remaining languages on Sep 10, 2020.

[32] For JA, RU, and ZH, the authors do not provide exact word counts. Therefore, we estimate them using other provided information (RU, ZH) or scripts for training corpus reconstruction (JA).

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 5 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 4 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), Figure 5 compares 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, 3, 4), 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 6b 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-MONOMODEL-MONOTOK model. Therefore, we plot the same correlation matrix excluding ID in Figure 3.

Compared to Figure 6b, 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 indeed 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 6a 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 1b and 1c).

3133

C Full Results

For compactness, we have only reported the performance of our models on the respective test datasets in the main text.[33] For completeness, we also include the full tables, including development (dev) dataset performance averaged over three random initializations, as described in §3. Table 8 shows the full results corresponding to Table 2 (initial results), Table 9 shows the full results corresponding to Table 3 (results for our new models), and Table 10 shows the full results corresponding to Table 4 (adapter-based training).

Lang	Model	Reference	V. Size	% Voc
MULTI	bert-base-multilingual-cased	Devlin et al. (2019)	119547	100
AR	aubmindlab/bert-base-arabertv01	Antoun et al. (2020)	64000	5.6
EN	bert-base-cased	Devlin et al. (2019)	28996	66.4
FI	TurkuNLP/bert-base-finnish-cased-v1	Virtanen et al. (2019)	50105	14.3
ID	indobenchmark/indobert-base-p2	Wilie et al. (2020)	30521	40.5
JA	cl-tohoku/bert-base-japanese-char	[5]	4000	99.1
KO	snunlp/KR-BERT-char16424	Lee et al. (2020)	16424	47.4
RU	DeepPavlov/rubert-base-cased	Kuratov and Arkhipov (2019)	119547	21.1
TR	dbmdz/bert-base-turkish-cased	Schweter (2020)	32000	23.0
ZH	bert-base-chinese	Devlin et al. (2019)	21128	79.4

Table 5: Selection of pretrained models used in our experiments. We display the respective vocabulary sizes and the proportion of tokens that are also covered by mBERT's vocabulary.

Lang	Treebank	# Words
AR	PADT	254192
EN	LinES, EWT, GUM, ParTUT	449977
FI	FTB, TDT	324680
ID	GSD	110141
JA	GSD	179571
KO	GSD	390369
RU	GSD, SynTagRus, Taiga	1130482
TR	IMST	47830
ZH	GSD, GSDSimp	222558

Table 6: UD v2.6 (Zeman et al., 2020) treebanks used for our tokenizer analyses. We use training and development portions only and display the total number of words per language.
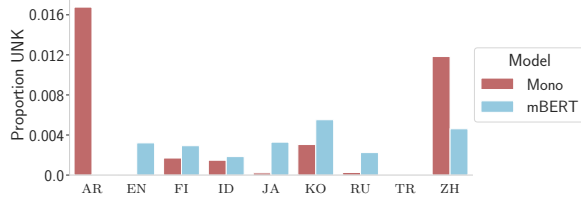

Figure 4: Proportion of unknown tokens in respective monolingual corpora tokenized by monolingual models vs. mBERT.

Task	Lang	Dataset	Reference	Train / Dev / Test
NER	AR	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	20000 / 10000 / 10000
	EN	CoNLL-2003	Tjong Kim Sang and De Meulder (2003)	14041 / 3250 / 3453
	FI	FiNER	Ruokolainen et al. (2020)	13497 / 986 / 3512
	ID	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	20000 / 10000 / 10000
	JA	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	20202 / 10100 / 10113
	KO	KMOU NER	[6]	23056 / 468 / 463
	RU	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	20000 / 10000 / 10000
	TR	WikiAnn	Pan et al. (2017); Rahimi et al. (2019)	20000 / 10000 / 10000
	ZH	Chinese Literature	Xu et al. (2017)	24270 / 1902 / 2844
SA	AR	HARD	Elnagar et al. (2018)	84558 / 10570 / 10570
	EN	IMDb Movie Reviews	Maas et al. (2011)	20000 / 5000 / 25000
	FI	—	—	—
	ID	Indonesian Prosa	Purwarianti and Crisdayanti (2019)	6853 / 763 / 409
	JA	Yahoo Movie Reviews	[7]	30545 / 3818 / 3819
	KO	NSMC	[8]	120000 / 30000 / 50000
	RU	RuReviews	Smetanin and Komarov (2019)	48000 / 6000 / 6000
	TR	Movie & Product Reviews	Demirtas and Pechenizkiy (2013)	13009 / 1627 / 1629
	ZH	ChnSentiCorp	[9]	9600 / 1200 / 1200
QA	AR	TyDiQA-GoldP	Clark et al. (2020)	14805 / 921
	EN	SQuAD v1.1	Rajpurkar et al. (2016)	87599 / 10570
	FI	TyDiQA-GoldP	Clark et al. (2020)	6855 / 782
	ID	TyDiQA-GoldP	Clark et al. (2020)	5702 / 565
	JA	—	—	—
	KO	KorQuAD 1.0	Lim et al. (2019)	60407 / 5774
	RU	SberQuAD	Efimov et al. (2020)	45328 / 5036
	TR	TQuAD	[10]	8308 / 892
	ZH	DRCD	Shao et al. (2019)	26936 / 3524
UD	AR	PADT	(Zeman et al., 2020)	6075 / 909 / 680
	EN	EWT	(Zeman et al., 2020)	12543 / 2002 / 2077
	FI	FTB	(Zeman et al., 2020)	14981 / 1875 / 1867
	ID	GSD	(Zeman et al., 2020)	4477 / 559 / 557
	JA	GSD	(Zeman et al., 2020)	7027 / 501 / 543
	KO	GSD	(Zeman et al., 2020)	4400 / 950 / 989
	RU	GSD	(Zeman et al., 2020)	3850 / 579 / 601
	TR	IMST	(Zeman et al., 2020)	3664 / 988 / 983
	ZH	GSD	(Zeman et al., 2020)	3997 / 500 / 500

Table 7: Named entity recognition (NER), sentiment analysis (SA), question answering (QA), and universal dependencies (UD) datasets used in our experiments and the number of examples in their respective training, development, and test portions. UD datasets were used for both universal dependency parsing and POS tagging experiments.

(a) AR   (b) EN   (c) FI

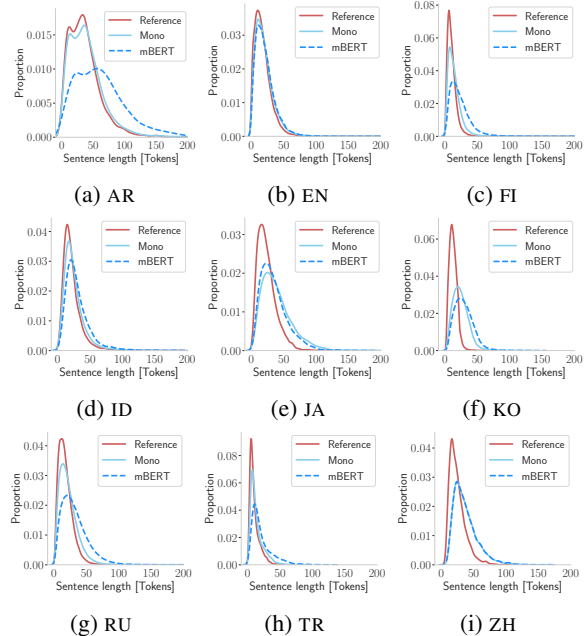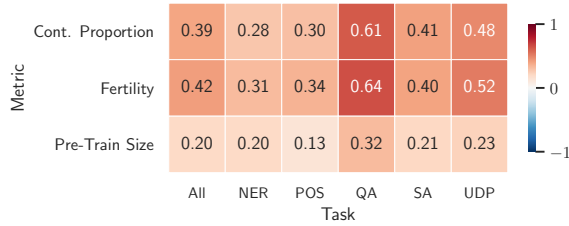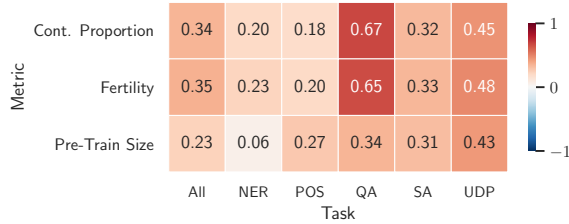(d) ID   (e) JA   (f) KO

(g) RU   (h) TR   (i) ZH

Figure 5: 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.

---

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

(a) We consider all languages and models.

(b) For the proportion of continued words and the fertility, we consider fully fine-tuned mBERT, the MONOMODEL-* models, and the MBERTMODEL-* models. For the pretraining corpus size, we consider the original monolingual models and the MONOMODEL-MONOTOK models.

Figure 6: 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.

Table 8: Full Results - Performance 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.

Lg	Model	NER Dev F_1	NER Test F_1	SA Dev Acc	SA Test Acc	QA Dev EM / F_1	UDP Dev UAS / LAS	UDP Test UAS / LAS	POS Dev Acc	POS 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
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
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
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
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
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
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
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
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
AVG	Monolingual	88.2	87.4	92.5	92.4	70.8 / 84.0	89.5 / 85.8	90.0 / 86.3	96.9	96.9
	mBERT	87.9	87.0	91.2	91.0	69.7 / 83.3	87.7 / 83.8	88.4 / 84.4	96.4	96.4

Table 9: Full Results - Performance of our new MONOMODEL-* and MBERTMODEL-* models (see §A.5) fine-tuned for the NER, SA, QA, UDP, and POS tasks (see §3.1), compared to the monolingual models from prior work and fully fine-tuned mBERT. We group model counterparts w.r.t. tokenizer choice to facilitate a direct comparison between respective counterparts. We use development sets only for QA. **Bold** denotes best score across all models for a given language and task. Underlined denotes best score compared to its respective counterpart.

Lg	Model	NER Dev F_1	NER Test F_1	SA Dev Acc	SA Test Acc	QA Dev EM / F_1	UDP Dev UAS / LAS	UDP Test UAS / LAS	POS Dev Acc	POS 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
	MONOMODEL-MONOTOK	88.6	**91.7**	96.0	95.6	67.7 / 81.6	88.4 / 83.7	89.2 / 84.4	97.3	96.6
	MONOMODEL-mBERTTOK	90.1	90.0	95.9	95.5	64.1 / 79.4	87.8 / 83.2	88.8 / 84.0	97.4	**97.0**
	mBERTMODEL-MONOTOK	**91.9**	91.2	95.9	95.4	66.9 / 81.8	88.2 / 83.5	89.3 / 84.5	97.2	96.4
	mBERTMODEL-mBERTTOK	90.0	89.7	95.8	95.6	66.3 / 80.7	87.8 / 83.0	89.1 / 84.2	97.3	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
FI	Monolingual	**93.3**	**92.0**	——	——	**69.9 / 81.6**	**95.7 / 93.9**	**95.9 / 94.4**	**98.1**	**98.4**
	MONOMODEL-MONOTOK	91.9	89.1	——	——	66.9 / 79.5	93.6 / 91.0	93.7 / 91.5	97.0	97.3
	MONOMODEL-mBERTTOK	91.8	90.0	——	——	65.1 / 77.0	93.1 / 90.6	93.6 / 91.5	96.2	96.7
	mBERTMODEL-MONOTOK	91.0	88.1	——	——	66.4 / 78.3	92.2 / 89.3	92.4 / 89.6	96.3	96.6
	mBERTMODEL-mBERTTOK	92.0	88.1	——	——	65.9 / 77.3	92.1 / 89.2	92.2 / 89.4	96.6	96.7
	mBERT	90.9	88.2	——	——	66.6 / 77.6	91.1 / 88.0	91.9 / 88.7	96.0	96.2
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
	MONOMODEL-MONOTOK	93.0	92.5	93.9	**96.0**	73.1 / 83.6	83.4 / 76.8	85.0 / 78.5	93.6	93.9
	MONOMODEL-mBERTTOK	93.3	93.2	93.9	94.8	67.0 / 79.2	84.0 / 77.4	84.9 / 78.6	93.4	93.6
	mBERTMODEL-MONOTOK	93.8	93.9	94.4	94.6	74.1 / 83.8	85.5 / 78.8	86.4 / 80.2	93.5	93.8
	mBERTMODEL-mBERTTOK	93.9	93.9	93.7	94.6	71.9 / 82.7	85.3 / 78.6	86.2 / 79.6	93.4	93.6
	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	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
	MONOMODEL-MONOTOK	87.9	87.1	89.0	88.8	72.8 / 90.3	87.9 / 84.2	89.8 / 86.6	96.4	96.7
	MONOMODEL-mBERTTOK	86.9	85.8	87.3	87.2	68.9 / 88.7	86.9 / 83.2	88.9 / 85.6	96.1	96.4
	mBERTMODEL-MONOTOK	87.9	86.6	88.2	88.1	72.9 / 90.2	87.9 / 83.9	90.1 / 87.0	96.2	96.5
	mBERTMODEL-mBERTTOK	86.7	86.2	86.6	86.6	69.3 / 89.3	87.2 / 83.3	89.2 / 85.9	95.9	96.2
	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
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
	MONOMODEL-MONOTOK	93.5	93.4	87.5	87.0	56.2 / 73.7	74.4 / 67.3	76.1 / 68.9	95.9	96.3
	MONOMODEL-mBERTTOK	93.2	93.3	85.8	84.8	55.3 / 72.5	73.2 / 66.0	75.3 / 68.3	96.4	96.5
	mBERTMODEL-MONOTOK	93.5	93.7	86.1	85.3	59.4 / 76.7	74.7 / 67.6	77.1 / 70.2	96.1	96.3
	mBERTMODEL-mBERTTOK	93.9	93.8	86.0	86.1	58.7 / 76.6	73.2 / 66.1	76.2 / 69.2	95.9	96.3
	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
AVG	Monolingual	91.5	91.1	92.5	92.6	68.0 / 82.3	87.2 / 82.4	88.3 / 83.7	96.2	96.2
	MONOMODEL-MONOTOK	91.0	90.8	91.6	91.9	67.3 / 81.7	85.5 / 80.6	86.8 / 82.0	96.0	96.2
	MONOMODEL-mBERTTOK	91.1	90.7	90.9	90.6	64.1 / 79.4	85.0 / 80.1	86.3 / 81.6	95.9	96.1
	mBERTMODEL-MONOTOK	91.6	90.7	91.2	90.9	68.0 / 82.2	85.7 / 80.6	87.1 / 82.3	95.9	95.9
	mBERTMODEL-mBERTTOK	91.3	90.3	90.5	90.7	66.4 / 81.3	85.1 / 80.0	86.6 / 81.7	95.8	95.9
	mBERT	91.2	90.4	90.5	90.0	66.3 / 81.2	84.7 / 79.6	86.1 / 81.0	95.6	95.6

Table 10: Full Results - Performance on the different tasks leveraging mBERT with different adapter components (see §4.4).

Lg	Model	NER Dev F_1	NER Test F_1	SA Dev Acc	SA Test Acc	QA Dev EM / F_1	UDP Dev UAS / LAS	UDP Test UAS / LAS	POS Dev Acc	POS Test Acc
AR	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^{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
	+ A^{Task} + A^{Lang}	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} + A^{Lang} + MONOTOK	91.5	91.1	96.0	95.7	67.7 / 82.1	87.7 / 82.8	88.5 / 83.4	97.3	96.5
FI	mBERT	90.9	88.2	——	——	66.6 / 77.6	91.1 / 88.0	91.9 / 88.7	96.0	96.2
	+ A^{Task}	91.2	88.5	——	——	65.2 / 77.3	90.2 / 86.3	90.8 / 87.0	95.8	95.7
	+ A^{Task} + A^{Lang}	91.6	88.4	——	——	65.7 / 77.1	91.1 / 87.7	91.8 / 88.5	96.3	96.6
	+ A^{Task} + A^{Lang} + MONOTOK	90.8	88.1	——	——	66.7 / 79.0	92.8 / 89.2	92.8 / 90.1	96.9	97.3
ID	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^{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
	+ A^{Task} + A^{Lang}	93.6	93.4	93.1	91.6	70.8 / 82.2	84.3 / 77.4	85.4 / 78.1	93.6	93.4
	+ A^{Task} + A^{Lang} + MONOTOK	93.0	93.4	94.5	93.8	74.4 / 84.4	84.6 / 77.6	85.1 / 78.3	93.7	93.5
KO	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^{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
	+ A^{Task} + A^{Lang}	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} + A^{Lang} + MONOTOK	87.7	86.5	87.9	87.9	73.1 / 90.4	87.0 / 82.7	88.9 / 85.2	96.3	96.5
TR	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^{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
	+ A^{Task} + A^{Lang}	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} + A^{Lang} + MONOTOK	92.7	92.7	86.1	85.3	60.0 / 77.0	73.5 / 65.6	75.7 / 68.1	96.4	96.3
AVG	mBERT	91.2	90.4	90.5	90.0	66.3 / 81.2	84.7 / 79.6	86.0 / 81.0	95.6	95.6
	+ A^{Task}	90.9	90.2	90.5	89.2	65.5 / 81.1	83.3 / 77.5	84.7 / 79.0	95.6	95.6
	+ A^{Task} + A^{Lang}	91.2	90.3	90.5	90.1	66.1 / 81.2	83.9 / 78.3	85.3 / 79.7	95.8	95.8
	+ A^{Task} + A^{Lang} + MONOTOK	91.1	90.4	91.1	90.7	68.4 / 82.6	85.1 / 79.7	86.2 / 81.0	96.1	96.0