# MicroBERT: Effective Training of Low-resource Monolingual BERTs through Parameter Reduction and Multitask Learning

**Luke Gessler     Amir Zeldes**
Department of Linguistics
Georgetown University
{lg876,amir.zeldes}@georgetown.edu

## Abstract

Transformer language models (TLMs) are critical for most NLP tasks, but they are difficult to create for low-resource languages because of how much pretraining data they require. In this work, we investigate two techniques for training monolingual TLMs in a low-resource setting: greatly reducing TLM size, and complementing the masked language modeling objective with two linguistically rich supervised tasks (part-of-speech tagging and dependency parsing). Results from 7 diverse languages indicate that our model, MicroBERT, is able to produce marked improvements in downstream task evaluations relative to a typical monolingual TLM pretraining approach. Specifically, we find that monolingual MicroBERT models achieve gains of up to 18% for parser LAS and 11% for NER F1 compared to a multilingual baseline, mBERT, while having less than 1% of its parameter count. We conclude reducing TLM parameter count and using labeled data for pretraining low-resource TLMs can yield large quality benefits and in some cases produce models that outperform multilingual approaches.

## 1 Introduction

Pretrained word embeddings are an essential ingredient for high performance on most NLP tasks. Transformer language models (TLMs)[1] such as BERT/mBERT (Devlin et al., 2019) RoBERTa/XLM-R, (Liu et al., 2019; Conneau et al., 2020), and ELECTRA (Clark et al., 2020) provide state-of-the-art performance, but they expect at least tens of millions of tokens in training data. High-resource languages like English, Arabic, and Mandarin are able to meet this requirement, but most of the world's languages cannot. Two major lines of work have arisen in order to address this

gap: the first attempts to use multilingual transfer to pool different languages' data together to meet TLMs' data demands, and the second attempts to lower TLMs' data demands by changing their architectures and training regimens.

In this study, we take up work in the latter direction, asking specifically whether (1) vast reduction of model size and (2) incorporation of explicitly supervised, rather than self-supervised, tasks into model pretraining can produce better monolingual TLMs. The former method is motivated by the intuition that normal-sized TLMs are so large as to be severely overparameterized for low-resource settings, and the latter method is motivated by an intuition that in the absence of large volumes of unlabeled text, signal from a supervised task with linguistic annotations is less likely be redundant to the model. We find evidence that indicates both methods are helpful: our MicroBERT models produce monolingual embeddings that can outperform comparable multilingual approaches. We summarize our contributions as follows:

1. We describe a method for training monolingual BERTs for low-resource settings, MicroBERT, characterized by a small parameter count and multitask pretraining which includes masked language modeling (MLM), part-of-speech (PoS) tagging and dependency syntax parsing.

2. Using evaluations on named-entity recognition (NER) and Universal Dependencies (UD) parsing across 7 diverse languages, we show that this approach is competitive with multilingual methods and often outperforms them for languages unseen by mBERT, even when the only pretraining task is MLM. Our evaluation reveals a 7% higher parser LAS and 6% higher NER F1 on average for unseen languages, with gains up to 18.87% and 11.81% for parsing and NER.

---

[1]Following popular usage, we will informally refer to TLMs similar to the original BERT as "BERTs" throughout this work.

3. We release all MicroBERT models trained for this work at `https://github.com/lgessler/microbert`.

4. We publicly release our code at `https://github.com/lgessler/microbert` for reproducing our results and as a turnkey facility for training new MicroBERTs.

## 2 Previous Work

At least since the development of pretrained static word embeddings (Mikolov et al., 2013b,a; Pennington et al., 2014; Bojanowski et al., 2017), pretrained word representations have been indispensable resources for NLP models, providing dense numerical representations of tokens' linguistic properties. Pretrained contextualized embeddings (McCann et al., 2018; Peters et al., 2018; Devlin et al., 2019) based on the Transformer architecture (Vaswani et al., 2017) have since overtaken them in popularity. Throughout this period, high-resource languages have received the majority of attention, and although interest in low-resource settings has increased in the past few years, there remains a large gap (in terms of linguistic resources, pretrained models, etc.) between low- and high-resource languages (Joshi et al., 2020).

### 2.1 Multilingual Models

The publication of BERT (Devlin et al., 2019) also included a multilingual model, mBERT, trained on 104 languages. mBERT and other massively multilingual models such as XLM-R (Conneau et al., 2020) achieve impressive performance not just on those 104 languages but also in some zero-shot settings (cf., *inter alia*, Pires et al. 2019; Rogers et al. 2020), despite the fact that models like mBERT do not have any explicit mechanism for inducing shared representations across languages. However, large language models like XLM-R suffer from the fact that languages necessarily compete for parameters, meaning that barring fortuitous synergies each additional language should tendentially degrade the overall performance of the model for a fixed parameter count. Moreover, languages with less training data tend to perform more poorly in LMs like XLM-R (Wu and Dredze, 2020).

While the majority of multilingual models seek to include many languages, with a large proportion of them being high-resource, there are some low-resource approaches to training multilingual models from scratch where there may not even be any high-resource languages. For example, Ogueji et al. (2021) train an mBERT on data totaling less than 1GB ($\approx$100M tokens) from 11 African languages, and find that their model often outperforms comparable massively multilingual models.

### 2.2 Adapting Multilingual Models

One response to the difficulties posed by massively multilingual models has been to leave aside the goal of fitting ever more languages into a single model, and to investigate whether it would be more fruitful to *adapt* pretrained massively multilingual models for a given target language. Enriching the TLM's vocabulary with additional tokens (e.g. wordpieces for BERT-style models) has been shown to be helpful because of how it improves tokenization and reduces the rate of out-of-vocabulary tokens (Wang et al., 2020; Artetxe et al., 2020; Chau et al., 2020; Ebrahimi and Kann, 2021). Transliteration has also been shown to be beneficial when there are related languages that would not have been able to benefit from transfer in the form of shared representations otherwise, e.g. between Turkish (Latin script) and related Uyghur (Arabic script) (Muller et al., 2021; Chau and Smith, 2021). Using adapter modules (Houlsby et al., 2019) has also proven effective (Pfeiffer et al., 2020a). All these approaches are typically combined with *continued pretraining*, where MLM and other pretraining tasks are used to update model weights, and some formulations of continued pretraining are multitask (Pfeiffer et al., 2020b; Chau and Smith, 2021, *inter alia).*

### 2.3 Monolingual Models

Whereas multilingual approaches have tried to address low-resource settings with transfer from high-resource languages, other approaches have investigated the question of how much data is needed for a given level of quality in a BERT-like model, and the question of what alternative training regimens might help reduce this data requirement.

Several studies have examined notable thresholds on dataset size. Martin et al. (2020) find in a series of experiments that for French, at least 4GB of text is needed for near-SOTA performance, and Micheli et al. (2020) show further that at least 100MB of text is needed (again for French) for "well-performing" models on some tasks. (Micallef et al., 2022) perform similar experiments for a monolingual Maltese BERT, finding that even when trained with only 46M tokens, the monolingual BERT, BERTu, was able to achieve results competi-

tive with an mBERT model adapted with the vocabulary augmentation methods of Chau et al. (2020). (Warstadt et al., 2020) train English RoBERTa models on datasets ranging from 1M to 1B tokens and find that while models acquire linguistic features readily on small datasets, they require more data to fully exploit these features in generalization on unseen data.

To our knowledge, there has been little work on examining whether significantly reducing model size could help in the low-resource monolingual setting. As a baseline, Chau and Smith (2021) and Muller et al. (2021) train monolingual BERTs with 6 instead of 12 layers for low-resource languages, but this does not even halve the model's parameter count. The only exception we were able to find is work from Turc et al. (2019), where very small models (as low as 4.4M parameters to BERT base's 110M) are pretrained directly prior to training via distillation, but the condition where the small model is only pretrained and not trained via distillation is not evaluated in their work.

## 2.4 Non-TLM Models

Finally, it is worth noting that while BERT-like TLMs are the clear winner overall for high-resource languages in most tasks, in low-resource settings, other embedding models may be superior. Arora et al. (2020) and Ortiz Suárez et al. (2020) show that ELMo (Peters et al., 2018), static (Pennington et al., 2014), and even random embeddings are often not too far behind BERT-like TLMs on some tasks even for high-resource languages. Riabi et al. (2021) show that a character-based language model is competitive with mBERT for one low-resource language, NArabizi.

## 3 Motivation

As we have seen, monolingual BERTs trained with standard methods tend to perform poorly when less than 20-40M tokens are available during training, and there is evidence that they do not learn to fully generalize some linguistic patterns without a large ($\approx$1B tokens, Warstadt et al. 2020) amount of training data. However, most popular methods for pretraining BERTs are self-supervised, using only unlabeled text. This has turned out well for high-resource languages, where unlabeled text is available in far greater quantities than labeled text, to the point where incorporating labeled text into pretraining does not always provide large gains.

However, even in very low-resource settings, it is common for sources of linguistic signal beyond unlabeled text to be available, such as treebanks, interlinearized text, and dictionaries. It is natural to ask whether using them as data for auxiliary supervised tasks during model pretraining could help monolingual models overcome a lack of unlabeled data, and perhaps even interact synergistically with the main pretraining task, such as MLM. It is known, for example, that BERTs learn to represent words' parts of speech (Rogers et al., 2020), and it seems possible that providing direct supervision for predicting parts of speech may help a model acquire good PoS representations with less data. This leads us to our first hypothesis **H1**, that monolingual models should benefit from multitask pretraining with auxiliary tasks incorporating labeled data.

Previous results also lead us to our second hypothesis **H2**, that in low-resource settings, monolingual BERTs are typically severely overparameterized. Most BERTs are overparameterized in the sense that they can have modules removed, disabled, or compressed while showing minimal regressions (or sometimes even improvements) (Rogers et al., 2020), but in H2 we mean further that there are so many parameters that the model cannot be effectively learned given the amount of data. As noted in §2, there appears to be a gap in the literature on whether pretraining a vastly scaled down BERT model could help monolingual BERTs perform better in low-resource settings, and we take up the question in this work.

## 4 Approach

We propose an architecture and training regime for monolingual BERTs which we call MicroBERT. We keep the basic architecture of BERT, but we reduce encoder layer count to 3, hidden representation size to 100, and number of attention heads to 5. (Compare this to BERT base's 12, 768, and 12, respectively.) Excluding prediction heads, this reduces parameter count from 108M[2] to 1.29M, or just 1.19% of a normal BERT model's size. After the encoder stack, one dedicated head is used for each task, where each head is provided with the last encoder layer's hidden states.

For training, assume a task set $\mathcal{T} = t_1, \ldots, t_{|\mathcal{T}|}$, corresponding datasets $\mathcal{D} = d_1, \ldots, d_{|\mathcal{T}|}$, and a set

---

[2]Obtained from `bert-base-cased` using the `BertModel` implementation in HuggingFace's `transformers` library.

of weights for each task $\lambda_1, \ldots, \lambda_{|\mathcal{T}|}$, s.t. $\sum_i \lambda_i = 1$. To prepare the sequences of batches $\mathcal{B} = b_1, \ldots, b_{|\mathcal{B}|}$ for a given epoch, construct each batch $b_i$ using only instances from exactly one dataset $d_t$, and sample batches so that each dataset $d_t$ is represented at least $\lfloor \lambda_t |\mathcal{B}| \rfloor$ times in $\mathcal{B}$. Each batch is sent not only to its dataset's corresponding prediction head, but also to any other prediction heads which are compatible with it. For example, a batch containing dependency syntax labels would be sent to the parsing prediction head, and it would also be sent to the MLM head, since the MLM head only requires unlabeled text.[3] If a dataset is exhausted in the course of this procedure, new instances are sampled anew from the beginning of the dataset. This is a simple means for addressing the fact that some datasets will be much larger than others, which without intervention could have led to one task's parameter updates drowning out others.

We consider three tasks in this work. The first is MLM implemented as whole-word, dynamic masking, as in RoBERTa (Liu et al., 2019). The second is PoS tagging, for which our prediction head is a simple linear projection. The third is dependency parsing, for which we use a modified form of the biaffine dependency parser of Dozat and Manning (2017) which has had the encoder LSTM stack removed. Cross-entropy loss is used for all tasks and summed together: each head produces an associated loss $\ell_i$, which is summed into a single loss $\ell$ which is used to begin backpropagation. We note that it would be straightforward to add other tasks, though we choose PoS tagging and parsing for this work since PoS tagged and dependency parsed datasets are relatively common for low-resource languages. This multitask setup is not novel—in fact, Chau and Smith (2021) use the the same three tasks for a similar purpose, though instead of pretraining a BERT from scratch, they use the multitask setup to perform adaptive finetuning on a pretrained multilingual model, and find a negative result.

## 5 Experimental Methods

To evaluate our approach, we train MicroBERT models on several languages and compare them to

---

| Language | Unlabeled | UD | NER |
|---|---|---|---|
| Wolof | 517,237 | 9,581 | 10,800 |
| Coptic | 970,642 | 48,632 | – |
| Tamil | 1,429,735 | 40,236 | 186,423 |
| Indonesian | 1,439,772 | 122,021 | 800,063 |
| Maltese | 2,113,223 | 44,162 | 15,850 |
| Uyghur | 2,401,445 | 44,258 | 17,095 |
| Anc. Greek | 9,058,227 | 213,999 | – |

Table 1: Token count for each dataset by language, sorted in order of increasing unlabeled token count. Recall that unlabeled data for Indonesian and Tamil was downsampled, and all other sources of unlabeled data were used in full.

a variety of baselines. All our experiments are implemented using AllenNLP (Gardner et al., 2018), Transformers (Wolf et al., 2020), and PyTorch (Paszke et al., 2019). All code and models are available at `https://github.com/lgessler/microbert`.

### 5.1 Data

We prepare datasets for seven diverse languages: Wolof, Uyghur, Ancient Greek, Maltese, Coptic, Indonesian, and Tamil. These languages were selected according to several criteria. First, two hard requirements were that they needed to have a Universal Dependencies (Nivre et al., 2016) treebank with a train, dev, and test split; and that they needed to have a "large-enough" source of unlabeled text totaling between 500,000 and 10,000,000 tokens. Second, languages were prioritized based on phylogenetic diversity: six unrelated language families are represented (Niger–Congo, Turkic, Indo-European, Afro-Asiatic, Austronesian, Dravidian), and languages vary widely in syntax (for example, Uyghur is morphologically rich, while Coptic is morphologically poor). Third, we sample languages along the spectrum of data quality—for example, some have very high quality tokenization, while others have noisier tokenization.

For each language, we obtain a UD treebank, a larger unlabeled corpus, and for all languages except Ancient Greek and Coptic, an NER dataset from WikiAnn (Pan et al., 2017). Unlabeled data for each language was taken from Wikipedia, except for Ancient Greek and Coptic, whose unlabeled corpora were taken from open access digital humanities projects. Note that the unlabeled corpora for Indonesian and Tamil were downsampled by randomly choosing Wikipedia articles until a quota of around 1.5M tokens was met. A summary of corpus statistics is given in Table 1, and

a full description of the languages' datasets and their preparation is given in Appendix B. Note that Uyghur is written in Arabic script; Wolof, Indonesian, and Maltese are written in Latin script; and Tamil, Coptic, and Ancient Greek are written in their own scripts.

## 5.2 Conditions

We compare four baselines, as well as six variants of the MicroBERT approach.

- **WORD2VEC**: a 100-dimensional static word embedding baseline, motivated by observations that static word embeddings can perform well in low resource settings (cf. §2).

- **MBERT**: the *bert-base-multilingual-cased* pretrained model. Note that only two of our seven languages (Indonesian and Tamil) have been seen by MBERT.

- **MBERT-VA**: the *bert-base-multilingual-cased* pretrained model adapted in the *vocabulary augmentation* method of Chau et al. (2020), where 99 wordpieces are added to the vocabulary and the model is pretrained further.

- **μBERT-M, μBERT-MX, μBERT-MXP**: our MicroBERT models with MLM; MLM and XPOS[4] tagging; and MLM, XPOS tagging, and UD parsing used in pretraining. μBERT-MX performs tasks at an 8:1 ratio, and μBERT-MXP performs tasks at an 8:1:1 ratio.

- **μBERT4-M, μBERT4-MX, μBERT4-MXP**: like the corresponding MicroBERT models, but approximately 4 times larger, having 200 instead of 100 hidden units; 8 instead of 5 attention heads; and 6 instead of 3 layers.

Our μBERT models are all trained for 200 epochs with a batch size of 32 and 8,000 batches per epoch, and we save the model that achieves best MLM performance on the validation split of the unlabeled dataset. This results in our models being trained on only 20% of the batches that BERT was, though we hypothesize that due to our smaller model and dataset sizes, this may not be an issue. A full description of our methods is given in Appendix C.

## 5.3 Evaluation

To evaluate our pretrained models, we perform NER on the WikiAnn datasets and dependency parsing on the UD datasets for each language–model pair, following previous work (Chau et al., 2020; Muller et al., 2021, *inter alia)*. We choose these tasks because they are common in the literature of TLM evaluation, because datasets are common even in low-resource languages for them, and because they both assess somewhat complementary linguistic information: informally, parsing requires grammatical knowledge, and NER requires semantic and world knowledge. Combined, they ought to give a holistic view of a model's abilities.

We use common hyperparameter settings to train the evaluation models which allow for fine-tuning of the BERT model at a reduced learning rate. A standard Dozat and Manning (2017) parser is used for the parsing evaluation, and a linear chain CRF with stacked LSTM encoders is used for the NER evaluation. Our metrics for these tasks are LAS and span-based F1 score respectively. Gold tokenization is used in both evaluations. No auxiliary input signals (e.g. PoS embeddings, morphological feature embeddings, static embeddings) are used. We forgo auxiliary inputs even though they would likely improve our scores, and even though it means no longer being able to compare our performance directly to numbers reported in some other works, since we believe providing the model's representations as the sole input provides the clearest picture of its quality.[5] Full descriptions of the evaluation models is available in Appendix D.

## 6 Results

Results for the parser evaluation are given in Table 2, and results for the NER evaluation are given in Table 3. For both tables, we also include additional rows comparing important model pairs.

It is possible to directly compare our parsing evaluation results with those of Chau and Smith (2021, Table 2), whose evaluation methodology we closely follow for parsing. For our three overlapping languages—Maltese, Uyghur, and Wolof—we

---

[4]In Universal Dependencies parlance, an XPOS tag is a part of speech tag from a language-specific tag inventory, as opposed to a UPOS, which is drawn from a universal tag inventory.

[5]This is motivated by our experience in preliminary experiments of using a parser with these auxiliary inputs, with the result that differences between our models were no larger than 3% since the auxiliary inputs were contributing so much to the model's performance, obscuring the content of the model representations. We also notice a similarly small difference between comparable models in other works where auxiliary inputs were used in a parsing evaluation.

| | Wolof | Coptic | Maltese | Uyghur | An. Gk. | Tamil | Indon. | Avg. |
|---|---|---|---|---|---|---|---|---|
| WORD2VEC | 72.35 | 85.69 | 73.41 | 54.27 | 73.30 | 50.91 | 74.10 | 69.15 |
| MBERT | 76.40 | 14.43 | 78.18 | 46.30 | 72.30 | **66.73** | 78.63 | 61.85 |
| MBERT-VA | 72.94 | 82.11 | 72.69 | 42.97 | 65.89 | 54.92 | 75.67 | 66.74 |
| Chau and Smith (2021) | 60.12 | | 65.92 | 60.34 | | | | |
| μBERT-M | 75.69 | 86.45 | 74.33 | 61.26 | 78.95 | 59.75 | 74.66 | 73.01 |
| μBERT-MX | **77.83** | **88.25** | 78.90 | **65.17** | **80.55** | 61.00 | 74.69 | **75.20** |
| μBERT-MXP | 73.30 | 86.35 | 75.11 | 59.98 | 79.08 | 58.05 | 73.28 | 72.16 |
| μBERT4-M | 74.42 | 82.72 | **79.25** | 57.79 | 79.59 | 61.09 | 74.32 | 72.74 |
| μBERT4-MX | 73.99 | 82.52 | 78.61 | 57.14 | 79.09 | 60.82 | 74.21 | 72.34 |
| μBERT4-MXP | 74.30 | 82.73 | 78.99 | 57.01 | 79.56 | 60.92 | 74.34 | 72.55 |
| μBERT-MX − MBERT-VA | 4.89 | 6.14 | 6.21 | 22.20 | 14.66 | 6.08 | -0.97 | 8.46 |

Table 2: Labeled attachment score (LAS) by language and model combination for UD parsing evaluation. The final row shows the difference in score between μBERT-MX and MBERT-VA. Results from Chau and Smith (2021)'s half-sized monolingual BERT are included for comparison.

| | Wolof | Maltese | Uyghur | Tamil | Indon. | Avg. |
|---|---|---|---|---|---|---|
| WORD2VEC | 86.89 | 82.67 | 86.37 | 82.71 | 94.28 | 86.58 |
| MBERT | 83.79 | 73.71 | 78.40 | 70.47 | 91.04 | 79.48 |
| MBERT-VA | 79.37 | 78.11 | 77.03 | 69.38 | 91.05 | 78.99 |
| μBERT-M | **83.92** | 75.89 | 81.36 | **82.28** | 92.25 | 83.14 |
| μBERT-MX | 81.12 | 84.80 | **85.45** | 81.61 | 92.43 | 85.08 |
| μBERT-MXP | 82.21 | **88.79** | 82.52 | 82.00 | 92.27 | **85.56** |
| μBERT4-M | 78.69 | 78.22 | 80.28 | 80.57 | **93.05** | 82.16 |
| μBERT4-MX | 80.95 | 80.00 | 79.36 | 80.12 | 92.55 | 82.60 |
| μBERT4-MXP | 79.02 | 79.31 | 81.59 | 80.11 | 93.01 | 82.61 |
| μBERT-MX − MBERT | -2.67 | 11.09 | 7.05 | 11.14 | 1.39 | 5.60 |

Table 3: Span-based F1 score by language and model combination for NER evaluation. The final row shows the difference in score between μBERT-MX and MBERT. Boldface indicating top performance for a language does not consider WORD2VEC.

find that LAS for mBERT is similar, which establishes that evaluation conditions are comparable. We include their half-size BERT model's numbers in Table 2 for comparison, which were obtained by training a `bert-base`-sized BERT from scratch on the target language with 6 instead of 12 layers.

**Non-DNN Baseline** First, corroborating prior work, we can see that static word embeddings are competitive for many languages, often outperforming the multilingual models in both tasks, and often performing best overall for NER.

**Multilingual Baselines** Note the generally poor performance of MBERT-VA, which we had hoped would be a baseline stronger than MBERT, but often underperforms relative to MBERT. An exception to this is parsing for Coptic, where MBERT's lack of wordpieces for Coptic script causes a high out-of-vocabulary rate, giving MBERT-VA an obvious advantage. After carefully ruling out implementation errors, we reason that MBERT-VA underperformed because fine-tuning a large BERT can produce unpredictable results (Rogers et al., 2020) and our hyperparameters for adaptive pretraining may have been suboptimal (Chau et al. 2020 perform a hyperparameter search for vocabulary augmentation—see Appendix D). In correspondence with the authors of Chau et al. (2020), we discussed our results, and they shared our assessment. In sum, MBERT-VA appears to produce volatile results without careful

hyperparameter selection, which we take to be a result of large model size and small dataset size.

**Monolingual Model Size** We can see that for parsing and NER, the μBERT4 model performs worse in almost all cases than the equivalent μBERT model. The degradation is -0.27% on average for -M variants, and -2.86% on average for -MX variants. The one language for which the μBERT4 model performs much better on parsing is Maltese, where the μBERT4-M model performs 5% better than the μBERT-M model, indicating that in this experimental condition greater model size may help, though note that the Chau and Smith's half-BERT does much worse than μBERT4-M showing a 13% lower score compared to μBERT-M and reversing the trend. On our two other languages in common with Chau and Smith, we see an 18% (Wolof) and 5% (Uyghur) degradation relative to μBERT-MX. For NER, we similarly observe that the μBERT4 variants have worse average performance than μBERT variants. We take this all to be strong evidence for H2, that monolingual BERTs trained at common sizes are severely overparameterized in low-resource settings, to the point that large performance degradations are observed.

**Parsing** Considering the five languages unseen by mBERT (all except Tamil and Indonesian), we see in the parsing results that in every case the best monolingual model, usually μBERT-MX, is

able to outperform the best multilingual model. In some cases the difference is very large, such as in Uyghur parsing where there is an absolute gain in 18.87% LAS, and in others it is within the range of chance, such as in Maltese parsing. For the languages mBERT has seen, Tamil and Indonesian, MBERT outperforms the μBERT by several points, though we find it remarkable that μBERT is able to still provide a competitive score despite being trained on very small subsets of Tamil and Indonesian Wikipedia (150K and 600K articles, respectively), which mBERT had full access to. μBERT-MX performs best of all the models, achieving a score 8.5% higher than that of MBERT-VA on average.

**NER**    Turning now to NER results, we see that in three cases, our μBERT models are able to clearly outperform other models, including Tamil, which MBERT has seen. In the other two cases, Indonesian and Wolof, μBERT models technically keep a lead but with margins thin enough to be noise. For all languages except Maltese however, WORD2VEC is able to meet or beat top performance from TLMs. Taken together with the parsing results, where WORD2VEC underperforms, and with the strengths and weaknesses of contextualized and static embeddings in mind, we hypothesize that NER on the WikiAnn dataset may require rote capacities, such as name memorization, instead of sophisticated linguistic knowledge, especially on an automatically-constructed dataset like WikiAnn.

**Validation MLM Perplexity**    In order to better understand the effects of our auxiliary tagging and parsing tasks, we examine the validation MLM perplexity of our models during pretraining. An example of these curves is given in Figure 1. We first observe that for all languages, validation MLM perplexity is lower at all times for the multitask models compared to the perplexity curve for the MLM only model. Moreover, validation MLM perplexity converges more quickly on its asymptotic value for multitask models. For μBERT-MX in particular, validation MLM perplexity usually comes very close to its final value even within the first 10 epochs of pretraining. Validation MLM perplexity is only one incomplete measure of model quality, and indeed it is not entirely predictive of downstream performance since μBERT-M sometimes outperforms μBERT-MX and μBERT-MXP. But we take these results as evidence that our auxiliary tasks
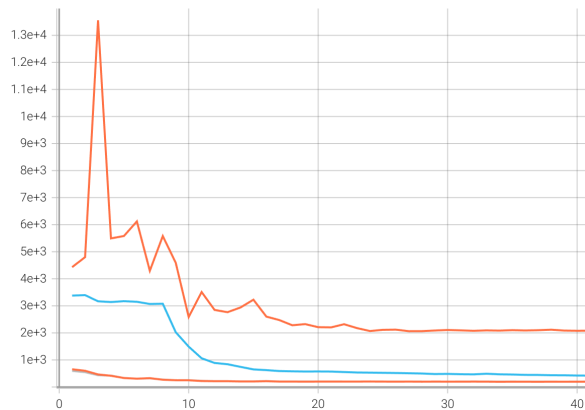


Figure 1: MLM perplexity vs. epoch for the validation split of the Uyghur dataset. The top line is μBERT-M, the middle line is μBERT-MXP, and the lowest line is μBERT-MX.

are helping our models learn more quickly. Moreover, while proving this would require additional work, it seems possible from the shapes of the validation curves that for the smallest datasets, multitask learning (MTL) might be helping models learn more than they could have through MLM alone.

Within validation MLM perplexity, we also see that each language follows one of two patterns: either the perplexity curves for μBERT-MX and μBERT-MXP are nearly identical, or the perplexity curve for μBERT-MXP remains a bit higher than for μBERT-MX.[6] With the intuition that more auxiliary tasks ought to make MLM easier, we had hypothesized that if anything the curve for μBERT-MX would have been higher than for μBERT-MXP, but instead the reverse sometimes turned out to be true. We hypothesize that the difference in task proportions between μBERT-MX and μBERT-MXP might have been partially responsible for this: in the former, 1 in 9 batches are for auxiliary tasks, and in the latter, 2 in 10 batches are for auxiliary tasks. If this is true, then finding the right proportion of primary and auxiliary tasks during pretraining would be critical for the multitask pretraining approach.

## 7    Discussion

**Main Findings**    We take our most important result to be our demonstration that it is possible to train a monolingual BERT from scratch that can compete with and even outperform multilingual models by up to 18% LAS and 11% NER F1 using

---

[6]The former pattern holds for Wolof, Maltese, Greek, Indonesian, and Tamil, and the latter pattern holds for Uyghur and Coptic.

as little as 500,000 tokens and a UD treebank of 44,000 tokens and less than 1% of the parameters.

**Multilingual Baselines**  We chose to use mBERT as a baseline because it is widely used and well studied. Moreover, given the the architectural homogeneity of mBERT and other leading multilingual LLMs, we additionally believe mBERT is strong enough to be representative of the state of multilingual LLMs for this work. While MBERT-VA appeared to severely underperform in some cases, we observe that it was still a strong baseline for both tasks (on average 5% better than MBERT for parsing, and 0.5% worse). In sum, while slightly stronger multilingual baselines may exist, we believe the ones in this work were still strong enough to show the MicroBERT approach holds promise, given that MicroBERTs were able to perform better than multilingual LLMs by several percentage points on average in both tasks.

**Hypotheses**  We find strong support in these results for H1, that monolingual TLMs often, though not always, benefit from multitask learning on labeled data in low-resource settings. We additionally find strong support for H2, that when data is severely limited, typical BERT configurations are harmfully overparameterized.

**Future Work**  There remain some unanswered questions in this work. The addition of the third parsing task proved harmful to performance in most cases, and it is unclear why. Parsing and XPOS tagging involve much of the same linguistic phenomena, and it seems possible that replacing one of them with a more semantic auxiliary task might have led to better results. Another possibility is that having loss computed for auxiliary tasks only on *some* batches may lead to jerky or suboptimal paths along the loss gradient, a problem which could be mitigated by having batches where only some sequences are suitable for use in auxiliary tasks.

It is natural to ask whether any of the elements of our approach here could find use in multilingual settings. Reducing the size of multilingual models may not be a promising direction due to the curse of multilinguality (Conneau et al., 2020). Ogueji et al. (2021) show further that even for low-resource multilingual models, size still seems to be important. As for multitask learning, Chau and Smith (2021) find a negative result for using MTL in multilingual model adaptation, though given the complex nature of MTL, many possible approaches remain untried.

Most languages in the world lack PoS tagged and parsed datasets, and if the MicroBERT approach is to be extended to very low-resource languages, it is likely that other auxiliary tasks would be needed. We leave this direction to future work, though we speculate that there are plenty of alternatives that may work. Parallel corpora, often in the form of a Bible translation, are readily available for over a thousand of the world's languages. High-quality rule-based morphological parsers are sometimes available for very low-resource languages, and their outputs could be used like PoS tags. Interlinearized texts and dictionaries are also common products of language documentation which are rich in linguistic information. All of these resources could be adapted for use in an auxiliary task.

## 8  Conclusion

We have shown that it is possible to train monolingual TLMs that are competitive with multilingual models using as little as 500K tokens and a 40K token treebank with greatly reduced model size and multitask learning on PoS tagging and dependency syntax parsing. While multilingual models did have some advantages over our approach, we observe that our MicroBERT approach has unique strengths for work on low-resource TLMs, including its lack of reliance on successful cross-lingual transfer and radically reduced computational demands for pretraining and downstream use.

We take this result to call into question whether multilingual representation learning can scale down effectively to truly "low-resource" languages that have less than a few million tokens in training data. Sometimes languages like these can be well served by transfer from related languages, even if all languages are low-resource (Ogueji et al., 2021), but not all languages may be so lucky: language isolates by definition lack related languages, and small language families are likely less able to benefit from transfer, since transfer tends to be enabled by phylogenetic (Nguyen and Chiang, 2017) or areal (Goyal et al., 2020) relatedness between languages. While multilingual methods hold much promise, it is important to examine other approaches to low-resource representation learning which, if not strictly better, may at least be complementary.

## References

Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. Contextual Embeddings: When Are

They Worth It? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2663, Online. Association for Computational Linguistics.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

Ethan C. Chau and Noah A. Smith. 2021. Specializing multilingual language models: An empirical study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.

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.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Abteen Ebrahimi and Katharina Kann. 2021. How to Adapt Your Pretrained Multilingual Model to 1600 Languages. *arXiv:2106.02124 [cs]*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. 2020. Contact relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 202–206, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. *arXiv:1902.00751 [cs, stat]*.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

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.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2018. Learned in Translation: Contextualized Word Vectors. *arXiv:1708.00107 [cs]*. ArXiv: 1708.00107.

Kurt Micallef, Albert Gatt, and Marc Tanti. 2022. Pre-training Data Quality and Quantity for a Low-Resource Language: New Corpus and BERT Models for Maltese. *arXiv*, page 12.

94

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*. ArXiv: 1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *arXiv:1310.4546 [cs, stat]*. ArXiv: 1310.4546.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When Being Unseen from mBERT is just the Beginning: Handling New Languages With Multilingual Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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).

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. 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, 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.

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.

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.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Arij Riabi, Benoît Sagot, and Djamé Seddah. 2021. Can Character-based Language Models Improve Downstream Task Performance in Low-Resource and Noisy Language Scenarios?

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv:2002.12327 [cs]*. ArXiv: 2002.12327 version: 3.

Caroline T. Schroeder and Amir Zeldes. 2016. Raiders of the Lost Corpus. *Digital Humanities Quarterly*, 010(2).

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. ArXiv:1908.08962 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *Advances in Neural Information Processing Systems*, 30.

A. Vatri and B. McGillivray. 2018. The Diorisis Ancient Greek Corpus: Linguistics and Literature. *Research Data Journal for the Humanities and Social Sciences*, 3(1):55 – 65. Place: Leiden, The Netherlands Publisher: Brill.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending Multilingual BERT to Low-Resource Languages. *arXiv:2004.13640 [cs]*. ArXiv: 2004.13640.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

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.

## A Acknowledgments

## B Datasets

| Treebank | Tokens |
|---|---|
| UD_Coptic-Scriptorium v2.9 | 48,632 |
| UD_Ancient_Greek-PROEIL v2.9 | 213,999 |
| UD_Indonesian-GSD v2.10 | 122,021 |
| UD_Maltese-MUDT v2.9 | 44,162 |
| UD_Uyghur-UDT v2.9 | 40,236 |
| UD_Wolof-WDT v2.9 | 44,258 |
| UD_Tamil-TTB v2.10 | 9,581 |

Table 4: Token count statistics for UD treebanks used in this work. Note that for this count, we count the constituent tokens of multiword tokens instead of counting a multiword token as a single token.

**Unlabeled** For Coptic, we use v4.2.0 of the Coptic SCRIPTORIUM corpora (Schroeder and Zeldes, 2016), obtained from https://github.com/copticscriptorium/corpora. For Ancient Greek, we use the initial release of the Diorisis corpus (Vatri and McGillivray, 2018), obtained from https://figshare.com/articles/dataset/The_Diorisis_Ancient_Greek_Corpus/6187256. Both corpora are preprocessed (tokenized, etc.) using language-specific tools to a quality higher than would have been obtained with a generic preprocessing pipeline. In Coptic's case, the data is further checked and with parts gold annotated by humans.

All other corpora are derived from Wikipedia. For Maltese, Uyghur, and Wolof, we use all available namespace 0 articles[7] as of February 2022, and for Indonesian and Tamil, we take a random sampling of namespace 0 articles as of June 2022, up to around 1.5M tokens.

All data is derived from Wikipedia's public dump files. While it is popular in NLP to use the text in the dump files directly, this is suboptimal, as the dump files' text contains markup, which makes the text noisy and means that document structural information cannot be used in the tokenization and sentence splitting process. We there-

---

[7]Wikipedia articles belonging to namespace 0 are main content articles instead of e.g. user pages or template pages.

---

fore take the additional step of rendering the dump into HTML using https://github.com/lgessler/wiki-thresher, which can then be used to obtain useful information about guaranteed sentence splits, e.g. between HTML elements like <p>. We perform rule-based sentence splitting and tokenization on this HTML to obtain our final tokenized texts.

For all 7 languages, we reserve around 10% of documents for validation and use the rest for training. A test split is unnecessary because our models are not being evaluated on unlabeled data.

**UD Treebanks** A summary of the treebanks we use and their versions is given in Table 4. We use the standard train/dev/test splits for all treebanks.

**WikiAnn Datasets** New train/dev/test splits were created in an 8:1:1 ratio for the WikiAnn dataset, which only divides sentences by language. It was not possible to split at the document level because no document metadata is available in the WikiAnn dataset. Tags are converted from the native IOB1 scheme into the BIOUL scheme. Some manual edits, logged in our version control history, were made to sentence boundaries in order to keep wordpiece sequence lengths below 512.

## C Conditions

All experiments for both pretraining and evaluation were performed on NVIDIA Tesla T4 GPUs with 16GB GDDR6 SDRAM.

**Word2vec** We use the Gensim (Rehurek and Sojka, 2011) implementation of the Word2vec skipgram with negative sampling algorithm for pretrained static word embeddings. The embeddings are trained just on the train split of the unlabeled corpus for each language. The vectors are 100-dimensional, window size is 5, and negative sampling factor is 5.

**mBERT-VA** We implement the Vocabulary Augmentation method exactly as prescribed by Chau et al. (2020) by training a new wordpiece tokenizer on the train split of the unlabeled data with a vocabulary size of 5,000, yielding a new monolingual vocabulary. The monolingual vocabulary is ranked by frequency of wordpieces, and the 99 unused tokens in mBERT's vocabulary indexed between 1 and 99 are replaced by tokens from the monolingual vocabulary which are not already present in mBERT's vocabulary. Since only preexisting token

indices are used, it is not necessary to modify the model's pretrained weights.

To train the weights of the previously unused token indices, adaptive pretraining with MLM is performed, again following Chau et al. (2020). Whereas Chau et al. perform a hyperparameter search, due to resource constraints we are forced to pick a single set of hyperparameters for adaptive pretraining, which we choose within the bounds of Chau et al.'s hyperparameter search. First, due to memory constraints on our GPUs, we are forced to set the batch size to 2. We pretrain for 20 epochs with 16,000 batches per epoch. The PyTorch AdamW optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.999$, learning rate at 1e-4, and weight decay at 0.05. The model which achieved lowest validation set perplexity is chosen.

**MicroBERT** Our tokenizer for MicroBERT is a WordPiece tokenizer. We scale vocabulary size from a minimum of 8,000 wordpieces up to 14,000 wordpieces, where the number of unique whitespace tokens for a given language determines how large the vocabulary will be. All models are uncased and perform Unicode NFD normalization as a preprocessing step during tokenization.

Since some tasks require wordpieces while others require tokens (e.g. PoS tagging), our encoder produces both wordpiece sequences and token sequences. The token sequence is constructed by keeping track of which wordpieces correspond to which original input tokens, and average pooling wordpieces for each token so that the sequence length reflects the number of original input tokens.

During data loading, sequences longer than 500 wordpieces are split into chunks of no more than 500 wordpieces each. Sequences this long only occur in the unlabeled datasets, so this does not pose a problem for producing valid losses on PoS tagging or parsing.

We train with a batch size of 32 for 200 epochs with 8,000 batches per epoch. We again use the AdamW optimizer with a learning rate of 3e-3, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay at 0.05. We allow early stopping if the validation metric, MLM perplexity, shows no improvement for 40 epochs. The model with the best validation MLM perplexity is selected.

While it is traditionally popular to train BERTs with triangular learning rates (Howard and Ruder, 2018), we chose not to use them for training our MicroBERTs. The reason is that, as noted by Raf-fel et al. (2020), it is necessary to know in advance approximately how many training steps are necessary to train a model, but since our MicroBERT architecture is much smaller, it is not obvious how many steps would be required to train it, making its use difficult. We do not expect this to lead to much worse performance compared to a properly configured triangular learning rate, as Raffel et al. (2020) also note that the triangular schedule often leads to only marginal gains compared to other schedules. Instead, we use PyTorch's ReduceL-ROnPlateau scheduler, which reduces learning rate when a certain number of validation steps have shown no improvement in MLM perplexity. We configure the scheduler so that if no improvement occurs for 2 epochs, the learning rate is halved, down to a minimum learning rate of 5e-5. Our results have shown that this training regimen can achieve good results, but we expect there is room for improvement and leave the task of refining it to future work.

## D Evaluation

**Parsing** We use the AllenNLP implementation of a biaffine attention parser (Dozat and Manning, 2017). In line with previous work, we set the dimensionality of the arc and tag representations to 100, and dropout and input dropout are set to 0.3. An encoder stack of 3 bidirectional LSTMs is used, with a recurrent dropout of 0.3, hidden size of 400, and highway connections. A scalar mix of representations from each layer of the BERT model is learned (Peters et al., 2018) to allow the model to fully exploit information present in earlier layers. Gold tokenization is used, and no supplementary representations (such as static word embeddings or feature or PoS embeddings) are provided.

We train for 300 epochs with a batch size of 16 and patience of 50 with LAS as our validation metric. To account for the very large size of some treebanks (e.g. Greek), we train for 200 batches per epoch. The AdamW optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate at 1e-3, and gradient clipping at 5.0. A reduced learning rate of 5e-5 is used for all parameters in the TLM.

**NER** We use AllenNLP's linear chain CRF tagger with BIOUL encoding. As with parsing, a scalar mix of representations from each layer of the BERT model is learned (Peters et al., 2018) to allow the model to fully exploit information present in earlier layers. An encoder stack of 2 bidirectional

LSTMs is used, with a dropout of 0.5 and hidden size of 200. The model's dropout is set to 0.5. Gold tokenization is used, and no supplementary representations (such as static word embeddings or feature or PoS embeddings) are provided.

We train for 300 epochs with a batch size of 16 and patience of 50 with span-based F1 as our validation metric. The AdamW optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate at 1e-3, and gradient clipping at 5.0. A reduced learning rate of 1e-5 is used for all parameters in the TLM.