Syntactic Inductive Bias in Transformer Language Models: Especially Helpful for Low-Resource Languages?

Luke Gessler **Nathan Schneider**
Department of Linguistics
Georgetown University
{lg876, nathan.schneider}@georgetown.edu

Abstract

A line of work on Transformer-based language models such as BERT has attempted to use syntactic inductive bias to enhance the pretraining process, on the theory that building syntactic structure into the training process should reduce the amount of data needed for training. But such methods are often tested for high-resource languages such as English. In this work, we investigate whether these methods can compensate for data sparseness in low-resource languages, hypothesizing that they ought to be more effective for low-resource languages. We experiment with five low-resource languages: Uyghur, Wolof, Maltese, Coptic, and Ancient Greek. We find that these syntactic inductive bias methods produce uneven results in low-resource settings, and provide surprisingly little benefit in most cases.

1 Introduction

Many NLP algorithms rely on high-quality pretrained word representations for good performance. Pretrained Transformer language models (TLMs) such as BERT/mBERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLM-R (Conneau et al., 2020), and ELECTRA (Clark et al., 2020) provide state-of-the-art word representations for many languages. However, these models require on the order of tens of millions of tokens of training data in order to achieve a minimum of quality (Micheli et al., 2020; Warstadt et al., 2020), a data requirement that most languages of the world cannot practically satisfy.

There are at least two basic approaches to addressing this issue. The first, which is at least as old as BERT, exploits multilingual transfer to reduce the data requirements for any individual language. The second aims to reduce TLMs' data requirements by modifying their architectures and algorithms. For example, Gessler and Zeldes (2022) more effectively train low-resource monolingual TLMs with as few as 500K tokens by reducing model size and adding supervised pretraining tasks with part-of-speech tags and syntactic parses.

We take up the latter direction in this work, looking specifically at whether the addition of syntactic inductive bias (SIB) during the pretraining procedure may help improve TLM quality in low-resource, monolingual settings. Specifically, we examine two methods which have been proposed for high-resource settings: the two syntactic contrastive loss functions of Zhang et al. (2022b), and the modified self-attention algorithm of Li et al. (2021), wherein a modified self-attention mechanism, restricted so that tokens may only attend to tokens that are syntactically "local", complements the standard self-attention mechanism.

At a high level, SIB is of interest in the context of TLMs because of how crucial self-attention is for TLMs' syntactic knowledge. In studies on an English TLM, BERT, Htut et al. (2019) and Clark et al. (2019) show that while syntactic relations are not directly recoverable from self-attention patterns, many self-attention heads seem to be sensitive to particular syntactic relations, such as that of a direct object or or a subject. But self-attention is *completely* unbounded: during pretraining, the model has to learn from scratch how to decide which other tokens in an input sequence a token should attend to. We therefore observe that if SIB could be effectively applied, then presumably self-attention weights would converge more quickly and learn more effectively, since their behavior has been observed to be so heavily syntactic in nature.

Moreover, we expect that this effect would be greater for low-resource languages, where the comparative lack of data is known to hamper models' ability to form robust linguistic representations. We find additional motivation for our interest in SIB given the nearly universal view held by linguists that the human mind does not start with the equivalent of a totally unconstrained self-attention

mechanism: for example, psycholinguists such as Hawkins (2014) have extensively documented processing-related constraints on syntax, and Generative linguists such as Ross (1967) have observed that many syntactic constructions which might have been possible are in fact not attested in English or any other language, and postulate that these constructions are at least in some cases "impossible" because of biologically-determined properties of the human mind. Our goal is therefore to give our models something like the constraints the human mind has in order to help them learn more effectively with less data.

We use a standard BERT-like TLM architecture as our base model, though we heavily reduce model size, following the results of Gessler and Zeldes (2022) which showed that this is beneficial in low-resource monolingual settings. We pretrain TLMs for five low-resource languages—Wolof, Coptic, Maltese, Uyghur, and Ancient Greek—varying which SIB methods are used. We then use Universal Dependencies (UD) (Nivre et al., 2016) syntactic parsing and WikiAnn (Pan et al., 2017) named entity recognition as representative downstream tasks that allow us to assess the quality of our models. Additionally, we evaluate our models using PrOnto (Gessler, 2023), a suite of downstream task datasets for low-resource languages. We find that these SIB methods are not very effective in low-resource languages, with small gains in some tasks and degradations or no effects in others. This is surprising given the intuition that SIB ought to help more in low-resource settings, and we speculate that other methods for SIB may be more effective in low-resource settings.

We summarize our contributions as follows:

1. We conduct what is, to the best of our knowledge, the first work examining whether SIB is helpful for pretraining low-resource Transformer LMs.
2. We reimplement SynCLM (Zhang et al., 2022b), SLA (Li et al., 2021), and MicroBERT (Gessler and Zeldes, 2022) in plain PyTorch and make it openly accessible.[1]
3. We present evidence from seven downstream evaluation tasks wherein the two SIB methods we examine are basically ineffective in our experimental settings, yielding only scattered and small gains.

[1]Our code is publicly available at https://github.com/lgessler/lr-sib .

2 Previous Work

Pretrained word representations have been essential ingredients for NLP models for at least a decade, beginning with static word embeddings such as word2vec (Mikolov et al., 2013b,a), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017). Contextualized word representations (McCann et al., 2018; Peters et al., 2018; Devlin et al., 2019) from Transformer-based (Vaswani et al., 2017) models have since overtaken them.

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 first modern multilingual TLM was mBERT, trained on 104 languages (Devlin et al., 2019). mBERT and other models that followed it, such as XLM-R (Conneau et al., 2020), demonstrated that multilingual pretrained TLMs are capable of good performance not on just languages represented in their training data, but also in some zero-shot settings (cf. Pires et al. 2019; Rogers et al. 2020, among others). But this is not without a cost: it has been shown (Conneau et al., 2020) that when a TLM is trained on multiple languages, the languages compete for parameter capacity in the TLM, which effectively places a limit on how many languages can be included in a multilingual model before performance significantly degrades for some or all of the model's languages. Indeed, the languages which had proportionally less training data in XLM-R's training set tended to perform more poorly (Wu and Dredze, 2020).

A possible solution to this difficulty is to *adapt* pretrained TLMs to a given target language, rather than trying to fit the target language into an ever-growing list of languages that the model is pretrained on. One popular method for doing this involves expanding the TLM's vocabulary with additional subword tokens (e.g. BPE tokens for RoBERTa-style models), which has been observed to improve tokenization and reduce out-of-vocabulary rates (Wang et al., 2020; Artetxe et al., 2020; Chau et al., 2020; Ebrahimi and Kann, 2021), leading to downstream improvements in model performance. But these and other approaches struggle

when a language is very far from any other language that a multilingual TLM was pretrained on.

Multilingual models like XLM-R which are trained on over 100 languages could be described as massively multilingual models. A more recent trend is to train multilingual models on just a few to a couple dozen languages, especially in low-resource settings. For example, Ogueji et al. (2021) train an mBERT on data drawn from 11 African languages, totaling only 100M tokens (cf. BERT's 3.3B), and find that their model outperforms massively multilingual models such as XLM-R, presumably because the African languages in question were quite unrelated to most of the languages XLM-R was trained on.

2.2 Monolingual Models

There has been comparatively little work exploring pretraining monolingual low-resource TLMs from scratch, and this lack of interest is likely explainable by the fact that monolingual TLMs require copious training data in order to be effective. Several studies have examined the threshold under which monolingual models significantly degrade, and all find that using standard methods, more data than is available in "low-resource" settings (definitionally, if we take "low-resource" to mean 'no more than 10M tokens') is required in order to effectively train a monolingual TLM. Martin et al. (2020) find at least 4GB of text is needed for near-SOTA performance in French, and Micheli et al. (2020) show further for French that at least 100MB of text is needed for "well-performing" models on some tasks. 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.

Gessler and Zeldes (2022) is the only work we are aware of which attempts to develop a method for training "low-resource" (<10M tokens in training data) monolingual TLMs. They extend the typical MLM pretraining process with multitask learning on part-of-speech tagging and UD syntactic parsing, and also radically reduce model size to 1% of BERT-base, yielding fair performance gains on two syntactic evaluation tasks. They find that their monolingual approach generally outperforms multilingual methods for languages that are not represented in the training set of a multilingual TLM (mBERT, in their study).

2.3 Syntactic Inductive Bias

Other work has investigated the syntactic capabilities of TLMs, and whether these capabilities could be enhanced with additional inductive bias. In an influential study, Hewitt and Manning (2019) find that structures that resemble undirected syntactic dependency graphs are recoverable from TLM hidden representations using a simple "structural probe", consisting of a learned linear transformation and a minimum spanning tree algorithm for determining tokens' syntactic dependents based on L2 distance. Kim et al. (2020) find similar results with a non-parametric, distance-based approach using both hidden representations and attention distributions. Both of these works attempt to find syntactic representations within a TLM without ever exposing a TLM to a human-devised representation. The quality of the recovered trees is usually poor relative to those obtainable from a syntactic parser, though their quality is consistently higher than random baselines.

Some works have attempted to provide models with direct access to human-devised representations—e.g., a syntactic parse provided in the Universal Dependencies formalism, which may have been produced by a human or by an automatic parser. Zhou et al. (2020) extend BERT by adding dependency and constituency parsing as additional supervised tasks during pretraining. Bai et al. (2021) assume that inputs are paired with parses, and use the parses to generate masks which restrict an ensemble of self-attention modules to attend only to syntactic children, parents, or siblings. Xu et al. (2021) use dependency parses to bias self-attention so that self-attention between tokens is weighted proportionally to the tokens' distance in the parse. In this paper, we examine the methods of Li et al. (2021) and Zhang et al. (2022b), which we describe below.

In sum, there are very many ways in which one could encourage a TLM to either learn a human representation of syntax, or to come up with (or reveal) its own. To our knowledge, none of the works on SIB have been examined in a low-resource TLM pretraining setting.

3 Approach

This work investigates whether methods for SIB that have succeeded in high-resource monolingual TLM pretraining settings could also be useful in analogous low-resource settings. As we have seen,

monolingual TLMs tend to have very poor quality when less than ≈10M tokens of training data are available for pretraining, and moreover, it has been observed that at least one dimension of this poor quality is models' inability to make grammatical generalizations without a large (≈1B tokens, Warstadt et al. 2020) pretraining dataset. Since it is (almost definitionally) difficult to get more data in low-resource settings, it is especially important to find other ways of improving model quality. It is therefore worthwhile to examine whether supplying some kind of SIB could help a low-resource TLM form better linguistic representations.

As discussed in §2.3, there are many ways to introduce SIB into a TLM. In this work, we look specifically at two methods: SynCLM (Zhang et al., 2022b) and SLA (Li et al., 2021), which is also used by Zhang et al. Li et al. (2021) extend the self-attention module with "local attention", wherein tokens may only attend to tokens which are $\leq k$ edges away in the dependency parse tree. Zhang et al. (2022b) devise two contrastive loss functions which are intended to encourage tokens to attend to sibling and child tokens, and in their experiments, they find success in combining these with SLA. A concise description of the details of each method is available in Appendix A.Both of these methods have only been evaluated on English, and both assume a UD syntactic parse as an additional input for each input sequence and use the parse in different ways to attempt to guide the model to better syntactic representations.

We use these two SIB methods with the model of Gessler and Zeldes (2022), MicroBERT, as a foundation. MicroBERT is a BERT-like model that has been scaled down to 1% of BERT-base, and that optionally employs part-of-speech tagging and syntactic parsing as auxiliary pretraining tasks. As shown by experiments on 7 low-resource languages conducted by Gessler and Zeldes (2022), MicroBERT performs much better than an unmodified BERT-base TLM, so we adopt it as our baseline model for most experiments in this work.

We now state our two main research questions:

- **(RQ1)** Do these SIB methods improve model quality when applied to a low-resource language?
- **(RQ2)** Are there any gains *complementary* with the part-of-speech tagging component of MicroBERT for training low-resource monolingual TLMs?

Language	Unlabeled	UD	NER
Wolof	517,237	9,581	10,800
Coptic	970,642	48,632	–
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 from Gessler and Zeldes (2022), sorted in order of increasing unlabeled token count.

4 Methods

4.1 Data and Evaluation

We reuse the datasets and evaluation setup of Gessler and Zeldes (2022), using five of their seven "truly"[2] low-resource languages' datasets. Each language's data includes a large collection of unlabeled pretraining data sourced from Wikipedia, as well as two datasets for downstream tasks for evaluation: UD treebanks for syntactic parsing, and WikiAnn (Pan et al., 2017) for named entity recognition (NER). We refer readers to Gessler and Zeldes' paper for further details on these datasets and the models for UD parsing and NER. In addition, we assess models on all five tasks in the PrOnto benchmark (Gessler, 2023), which will be described below.

4.2 Models

We reimplement the MicroBERT model of Gessler and Zeldes (2022), as well as the work of Zhang et al. (2022b) and Li et al. (2021). In all cases, we reuse code wherever possible and closely check implementation details and behavior in order to ensure correctness. As a foundation, we use the BERT implementation provided in HuggingFace's `transformers` package (Wolf et al., 2020), and we also use AI2 Tango[3] for running experiments. We obtain all of our parses for the unlabeled portions of our datasets automatically using Stanza (Qi et al., 2020), following Zhang et al.

In order to answer our research questions, for each language, we examine the following conditions:

1. MBERT – plain multilingual BERT (`bert-base-multilingual-cased`). A baseline; numbers taken from Gessler and Zeldes.

[2]The Indonesian and Tamil Wikipedias were larger than Gessler and Zeldes' cutoff of 10M tokens for "low resource", and Indonesian and Tamil are also included in mBERT's pretraining data. We exclude them for the purposes of this study in the interest of examining these five truly low-resource languages in more depth.

[3]https://github.com/allenai/tango

Model	Wolof	Coptic	Maltese	Uyghur	An. Gk.	Avg.
MBERT	76.40	14.43	78.18	46.30	72.30	57.52
MBERT-VA	72.94	82.11	72.69	42.97	65.89	67.32
µB-M	77.71	88.47	81.40	59.97	81.94	77.90
µB-MP	75.88	87.90	80.88	59.42	81.15	77.05
µB-MT	77.29	88.32	81.06	59.79	81.42	77.58
µB-MPT	77.05	88.38	80.07	58.94	81.35	77.16
µB-MPT-SLA	76.25	87.87	79.52	58.37	80.77	76.56
µB-MX	77.74	88.00	81.25	61.23	82.02	78.05
µB-MXP	77.90	88.63	82.21	60.62	81.34	78.14
µB-MXT	77.30	88.34	81.87	60.44	82.11	78.01
µB-MXPT	78.19	88.48	81.30	61.41	81.80	78.24
µB-MXPT-SLA	76.89	87.90	80.87	59.35	81.17	77.24

Table 2: Labeled attachment score (LAS) by language and model combination for UD parsing evaluation. Results for MBERT and MBERT-VA are taken from Gessler and Zeldes (2022).

Model	Wolof	Maltese	Uyghur	Avg.
MBERT	83.79	73.71	78.40	78.63
MBERT-VA	79.37	78.11	77.03	78.17
µB-M	83.40	82.98	86.70	84.36
µB-MP	86.38	84.16	87.44	86.00
µB-MT	87.16	89.46	87.33	87.98
µB-MPT	88.89	86.83	87.67	87.80
µB-MPT-SLA	86.38	84.85	84.81	85.35
µB-MX	77.65	86.09	89.75	84.49
µB-MXP	81.45	87.74	87.41	85.54
µB-MXT	85.94	84.67	87.98	86.19
µB-MXPT	87.06	84.37	87.53	86.32
µB-MXPT-SLA	83.72	85.35	88.07	85.71

Table 3: Span-based F1 score by language and model combination for NER evaluation.

2. MBERT-VA – MBERT, but with vocabulary augmentation. A baseline; numbers taken from Gessler and Zeldes.
3. µB-M – plain MicroBERT trained only using MLM. We obtain our own numbers to verify the correctness of our implementation.
4. µB-MP, µB-MT, µB-MPT – MicroBERT with either one or both of the SynCLM loss functions: P indicates the phrase-guided loss, and T indicates the tree-guided loss.
5. µB-MPT-SLA – µB-MPT, with the addition of SLA. We follow Zhang (2022) in using SLA only in conjunction with both contrastive losses.
6. µB-MX, µB-MXP, µB-MXT, µB-MXPT, µB-MXPT-SLA– the conditions in (3–5), but with the addition of part-of-speech tagging (X) as an auxiliary pretraining task. This is done using the same methods of Gessler and Zeldes: PoS tagging is only performed on gold-tagged data from the UD treebank, and tagged sequences are mixed into the pretraining data at a 1 to 8 ratio.

Revisiting our research questions, we intend for the conditions in (3–5) to provide evidence for **(RQ1)**, and for the additional information from the conditions in (6) to provide evidence for **(RQ2)**.

5 Results

Parsing Our results for UD syntactic parsing are given in Table 2. While all models beat the multilingual baselines, neither SynCLM nor SLA seems to improve model quality. In the -M variant models, the top-performing model is always the one trained with plain masked language modeling. This is not so for the -MX variant models, where the -MXP and -MXPT models do slightly better on average, though this difference is small enough to be within the range of experimental noise. Surprisingly, -MPT-SLA models do worst of all. Finally, comparing -M variants to their -MX counterparts, we do find that in all cases the -MX counterpart is better on average, and that the difference is about 1% LAS.

NER Our results for WikiAnn NER are given in Table 3. Considering the -M variant models first, we see that in all cases the model trained using only MLM performs the worst, and the -MPT-SLA variant, while always no better than the -MP, -MT, and -MPT variants, also outperforms the plain MLM model. The -MP, -MT, and -MPT variants do best with a difference of up to 4 points F1 on average.

Turning now to the -MX variants, while it is still true that on average the plain MLM model performs worst and the non-SLA SynCLM models perform best, there is more variation within individual languages. The best model for Uyghur is the plain MLM model, and for Maltese, the plain MLM model outperforms µB-MXT and µB-MXPT.

Considering now all the NER results, two patterns are worth noticing. First, unlike in parsing, a -MX variant does not always outperform its -M counterpart: for example, µB-MP for Wolof is better than µB-MXP by a difference of 5 points F1. We can see further that the -M models beat the -MX

Model	Non-pronominal Mention Count					Same Sense					All 5
	An. Grk.	Coptic	Uyghur	Wolof	Avg.	An. Grk.	Coptic	Uyghur	Wolof	Avg.	Avg.
µB-M*	52.59	50.75	49.37	51.47	51.04	60.58	61.32	60.65	59.78	60.58	68.65
µB-MX*	56.81	53.34	51.19	59.24	55.14	60.95	61.30	61.51	63.08	61.71	70.01
MBERT	57.36	49.52	51.46	57.35	53.92	65.34	52.79	62.73	66.49	61.84	67.92
µB-M	56.68	52.52	52.72	53.78	53.93	**58.51**	56.65	57.97	58.54	57.92	68.25
µB-MP	56.13	51.98	**54.39**	**54.41**	54.23	58.41	**58.15**	59.54	**58.95**	**58.76**	**68.40**
µB-MT	50.41	48.98	49.37	51.47	50.06	58.48	58.08	57.99	57.03	57.90	66.88
µB-MPT	53.68	48.98	51.74	51.47	51.47	53.36	54.19	59.32	58.07	56.23	66.39
µB-MX	**57.49**	53.07	**54.39**	53.57	**54.63**	56.71	56.01	58.88	58.18	57.44	68.39
µB-MXP	54.09	**53.34**	**54.39**	53.78	53.90	55.61	55.02	59.47	58.47	57.14	67.84
µB-MXT	53.95	51.02	49.37	51.47	51.45	57.44	56.37	**59.56**	57.93	57.83	66.89
µB-MXPT	52.72	51.71	50.91	51.47	51.70	57.19	56.17	56.81	58.14	57.08	67.30

Table 4: Accuracy by language and model combination for two tasks in PrOnto: the Non-pronominal Mention Count, and Same Sense tasks. For non-baseline models, an underline indicates the best performance for a language–task combination for a particular model variant (-M or -MX), and boldface indicates the best performance across either model variant. Scores for MBERT, µB-M*, and µB-MX* are taken from Gessler (2023)—the asterisk indicates that the latter two models are not our implementation but the one provided in Gessler and Zeldes (2022), which is reported in Gessler (2023). Rightmost column contains an average over all languages and tasks for a given model. Results for PrOnto's other three tasks are given in Appendix D.

models on average by about 4 points F1. This indicates that when combined with SLA and SynCLM, the PoS tagging pretraining task does not appear to be helpful for dimensions of model quality that are implicated in NER. Second, the addition of -SLA never results in a gain relative to any of the SynCLM models, except for Uyghur, where it produces a gain of 0.09, which is within the range of experimental noise.

PrOnto We run our SynCLM models[4] on all five tasks of PrOnto (Gessler, 2023) on all languages except Maltese, which is not represented in PrOnto because of the lack of an open-access Maltese Bible. For each language in PrOnto, a dataset for five sequence classification tasks is available which was constructed by aligning New Testament verses from the target language with the English verse in OntoNotes (Hovy et al., 2006) and projecting annotations from English to the target language. All 5 tasks are sequence classification tasks. Each task requires a model to predict a certain grammatical or semantic property—these are, respectively: the number of referential noun phrases in a sequence; whether the subject of a sentence contains a proper noun; the sentential mood of a sentence; whether two input sequences both contain a usage of a verb sense; and whether two input sequences both contain a usage of a verb sense with the same number of arguments. We refer readers to the PrOnto publication for further details.

Results from two of the five tasks are given in Table 4.[5] Broadly, we may observe that the -MPT and -MXPT models never perform best within a language, with either variant being in many cases worse by a few absolute points compared to other models. Looking at -M-family models, -MP is the clear winner, doing a little better than -M and much better than -MT or -MPT on both tasks. By contrast, for -MX-family models, the -MXP variant does a bit worse on average than -MX, and for the Same Sense task, the -MXT model does a bit better than -MXP. Looking to the rightmost column in Table 4, we can see that when we average accuracy scores for a model across all languages and all 5 tasks in PrOnto, the -MP model has the highest score overall, with -MX and -M very close behind and all other model variants quite a ways behind.

Overall, it seems that for the PrOnto tasks, of all the syntactic bias methods we have tried, only the use of the phrase-based contrastive loss (-MP) or the tree-based contrastive loss in combination with PoS tagging (-MXT) showed much improvement over the baselines. In individual language–task combinations, models sometimes had multiple-point performance differences over others, but when considered in aggregate, only -MP shows any improvement over -M and -MX—by 0.15% and 0.01% accuracy, respectively.

6 Discussion

Considering first whether SynCLM and SLA yield benefits for low-resource monolingual TLMs

[4] It was not possible to run our SLA models on PrOnto due to considerable implementation effort that would have been required, so we omit those models from this evaluation.

[5] We omit results from the other 3 from the main body for space reasons—see Appendix D for these results.

(RQ1), we have found positive evidence from the WikiAnn NER experiments, and weak positive evidence from the PrOnto experiments. It is true that the same methods did not produce measurable gain for the UD parsing task, but this is in line with previous findings for these two methods, where on some downstream evaluations, gain was very small or slightly negative—we return to this matter in the following paragraph. For the question of whether these benefits are complementary with the PoS tagging pretraining strategy introduced in Gessler and Zeldes (2022) (RQ2), we do not find consistent evidence in any of our experiments that both PoS tagging and SynCLM or SLA yield complementary benefits. The only positive evidence we find for this is in the PrOnto experiments, where the -MXT model variant does better than -MX in some task–language combinations, though worse overall.

The difference in the way model variants behaved in these seven evaluation tasks is striking, and it is difficult to understand why models exhibited these different behaviors. It is worth comparing these results with those reported by the SynCLM authors (Zhang et al., 2022b). For many of the GLUE tasks that they assess their models on (their Table 3), there is little or no improvement from adding -P, -T, or -PT-SLA. For example, considering their models based on RoBERTa-base, none of their model variants outperform the MLM-only baseline for the QQP (Quora Question Pairs2), STS (Semantic Textual Similarity), or MNLI-m (Multi-Genre Natural Language Inference, matched). This situation is more or less analogous to the one we observed in our experiments for the UD parsing downstream task, where the addition of SynCLM and SLA had basically no effect.

On the other hand, the GLUE task with the greatest gain, CoLA (Corpus of Linguistic Acceptability), shows a difference of only 1.7% Matthews correlation coefficient, and a couple of other tasks like SST (Stanford Sentiment Treebank), show an improvement of only 0.3% accuracy. It would be naïve to directly compare percentage points of different metrics in totally different experimental settings and make conclusions about effect sizes, we nevertheless point out that we observe improvements of 1–4% F1 in our NER experiments for -M models. In light of this, we consider our results to be broadly in line with the trend for previous works' results on English: there is no improvement that is wholly consistent across evaluations, and only modest gains for the benchmarks that do improve.

In summary, we find that SynCLM and SLA produce uneven results in low-resource settings, though we also find that when they do succeed, they can yield gains that appear greater than anything observed for high-resource languages: we saw that when we take a pure MLM pretraining regimen as a base and add SynCLM and/or SLA, we are able to improve the quality of pretrained TLMs by 1 to 4 absolute points F1 in NER. While a similar benefit was not observed for UD parsing, it is also true that there was a noticeable degradation on UD parsing in only a couple cases, and in most cases simply had no effect.

7 English Experiments

One might have expected SIB to be a knockout success for low-resource languages given the intuitive feeling that at lower data volumes, additional bias ought to be more helpful. We considered reasons why our attempts to do this might not have panned out—perhaps, for example, tree structure matters most for highly analytic languages like English, or perhaps the tasks used to evaluate English in GLUE are more sensitive to high-level sentence structure, or perhaps sensitivity to syntax is only advantageous given a base model with sufficiently rich distributional information. Here, we consider another possible explanation: that the inductive bias with these methods only helps given high-quality syntactic parses. An obvious difference between English and the languages we have examined in this study is that UD parsers for English generally achieve much higher performance given the size and annotation quality of English UD treebanks. This is a potentially consequential difference, given that both the SynCLM and SLA methods rely on UD parse trees as inputs. In addition, the models we have developed here differ from common kinds of English BERTs in that they are much smaller and were trained on much less data, and it is possible that the SynCLM and SLA methods might have interactions with these two variables of model construction.

In order to investigate whether parse tree quality, model size, and pretraining data size might be consequential for these SIB methods, we run several additional experiments on English datasets. We choose English because its status as a high-resource language allows us control over several

independent variables which we do not have control over in low-resource settings, namely data quantity, syntactic parse quality, and model size.[6] We can frame an additional research question that we wish to answer:

- **(RQ3)** Are SynCLM and SLA sensitive to parse tree quality, model size, or pretraining dataset size?

For our English dataset, we use AMALGUM (Gessler et al., 2020) as our source of pretraining data. AMALGUM contains around 2M tokens and contains automatic parses with quality that exceeds what can normally be obtained from a standard parser. For downstream evaluation, we use the English Web Treebank (Silveira et al., 2014), which contains around 250K tokens, and the English split of WikiAnn, downsampled to around 50K tokens in order to bring it closer to the quantities for our other 3 languages (cf. Table 1). In addition, we use a 100M subset of BERT's pretraining data as a larger source of unlabeled pretraining data.

We frame these additional conditions for English, extending our model naming scheme from above:
1. -NP – syntax trees are taken from Stanza in the same way as before.
2. -HQP – syntax trees are taken from AMALGUM's annotations, made by a **h**igh **q**uality **p**arser.
3. -BD – pretraining is done using the **b**ig **d**ataset instead of AMALGUM.
4. -BD-BM – like -BD, and in addition, the model size is set to half of BERT-base (6 layers instead of 12).

Evidence from these conditions could tell us more about how and when SynCLM and SLA can succeed in low-resource scenarios. We pretrain these models as we did in our main experiments and evaluate them on UD parsing and WikiAnn NER.

A full description of our results is given in Appendix B, and we give a description of our key finding here: that SynCLM and SLA are not very sensitive to parse quality or model size, but are sensitive to quantity of pretraining data. The insensitivity to parse quality may come as a surprise, and we reason that this is actually understandable, since both methods focus mostly on low-height subtrees (often corresponding to phrase- or sub-phrase-level constituents) which are more likely to be correct even when overall parse quality is bad. We find evidence for sensitivity to data size in the fact that SynCLM and SLA provide gains of up to 1% F1 for the NER evaluation in the two low-data conditions, while in the higher-data conditions, all but one of the bias-enhanced models lead to degradations relative to the baseline. In sum, we take this to show that lower parse quality is not the major reason for the ineffectiveness of SynCLM and SLA in low-resource settings.

[6]Model size is not controllable in low-resource settings in the sense that, as Gessler and Zeldes (2022) argued, monolingual low-resource TLMs exhibit severe degradations when they get too large.

8 Conclusion

In this work, we have taken two methods for SIB that have succeeded in English, SynCLM and SLA, and we have investigated whether they may also be beneficial in low-resource monolingual settings. We find that in most cases these methods do not result in an improvement in model quality as measured on seven tasks. Further, in our auxiliary experiments on English, we found evidence suggesting that the lower quality of parses in low-resource settings is probably not what is driving the ineffectiveness of these SIB methods.

Considering all of our results, we conclude that these two specific methods—SynCLM and SLA—are not well suited to supporting the pretraining of language models in low-resource settings, but we also view it as a yet open question whether any method for SIB could succeed in this role. There are some reasons why SynCLM and SLA might have been unhelpful. First of all, recall the fact that SynCLM limits its application to only short subtrees (no taller than 3 nodes). This would mean that most of the time, the contrastive loss functions would only be operating on basic phrase-level constituents, such as noun phrases, and not higher, clause-level phenomena such as relations between the main clause's predicate and its arguments. If it were the case that the former kind of syntax is relatively easy for models to learn even with limited data, and that the latter kind of syntax is what is hard and therefore where SIB really ought to help, then we would expect to see the results we found in this work, where neither method did much to help.

Therefore, while we find little reason to be optimistic about these two particular methods in low-resource settings, we don't view the evidence in this paper as an indictment of SIB in low-resource settings in general, and suggest that SIB methods which are better able to provide bias for higher, clause-level syntactic dependencies may produce better results for low-resource languages.

Acknowledgments

We thank Amir Zeldes for very helpful comments on this work.

References

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.

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or Annotate? Domain Adaptation with a Constrained Budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5002–5015, Online and Punta Cana, Dominican Republic. 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.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. 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.

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

Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860.

Luke Gessler. 2023. Pronto: Language model evaluations for 859 languages.

Luke Gessler, Siyao Peng, Yang Liu, Yilun Zhu, Shabnam Behzad, and Amir Zeldes. 2020. AMALGUM – a free, balanced, multilayer English web corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5267–5275, Marseille, France. European Language Resources Association.

Luke Gessler and Amir Zeldes. 2022. MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning. In *Proceedings of the The 2nd Workshop on Multilingual Representation Learning (MRL)*, pages 86–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

John A. Hawkins. 2014. *Cross-Linguistic Variation and Efficiency*. Oxford University Press. Publication Title: Cross-Linguistic Variation and Efficiency.

John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, New York, New York. Association for Computational Linguistics.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in BERT track syntactic dependencies? *CoRR*, abs/1911.12246.

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.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2021. Improving BERT with syntax-aware local attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, 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.

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.

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.

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.

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.

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.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

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.

John Robert Ross. 1967. *Constraints on Variables in Syntax*. Doctoral Dissertation, Massachusetts Institute of Technology.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

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.

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, 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. 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.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422, Online. Association for Computational Linguistics.

Bryan Zhang. 2022. Improve MT for search with selected translation memory using search signals. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 123–131, Orlando, USA. Association for Machine Translation in the Americas.

Rui Zhang, Yangfeng Ji, Yue Zhang, and Rebecca J. Passonneau. 2022a. Contrastive data and learning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 39–47, Seattle, United States. Association for Computational Linguistics.

Shuai Zhang, Wang Lijie, Xinyan Xiao, and Hua Wu. 2022b. Syntax-guided contrastive learning for pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2430–2440, Dublin, Ireland. Association for Computational Linguistics.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT : Linguistics Informed Multi-Task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

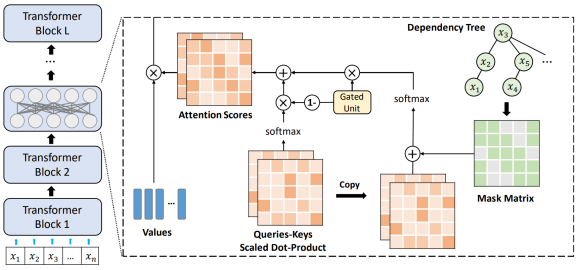

Figure 1: Figure 1 from Li et al. (2021). The standard self-attention mechanism is complemented by another self-attention mechanism in which tokens may only attend to tokens close to it in a parse tree. A gated unit with learnable parameters interpolates the two attention distributions before the distribution is combined with the Value representation.

A Summary of SLA and SynCLM

Our approach critically relies on two previous results, which we summarize here.

A.1 Syntax-aware Local Attention

Li et al. (2021) introduce Syntax-aware Local Attention (SLA), a variation on a standard TLM self-attention mechanism that retains standard self-attention and complements it with a separate self-attention mechanism where each token may only attend to "syntactically local" tokens.

Recall that BERT and most other TLMs use scaled dot-product attention in every attention head, where the attention distribution $\mathbf{A}$ can be computed with query and key representations $\mathbf{Q}$ and $\mathbf{K}$, d is the size of an individual attention head's hidden representation, and the attention head's output $\mathbf{O}$ is the product of $\mathbf{A}$ and the value representation $\mathbf{V}$:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \tag{1}$$

$$\mathbf{O} = \mathbf{A}\mathbf{V} \tag{2}$$

Now, assume an input sequence $W = w_1, \ldots, w_n$ with an unlabeled dependency parse $H = h_1, \ldots, h_n$ where h_i indexes token w_i's syntactic head. Define syntactic distance between two words, $D(w_i, w_j)$, as the length of the shortest path between the two words in the parse:

$$D(w_i, w_j) \coloneqq \textsc{Shortest-Path}(H, i, j) \tag{3}$$

To account for the fact that parses may be inaccurate (e.g. if they come from an automatic parser), define *windowed* syntactic distance like so:[7]

$$D'(w_i, w_j) = \min_{k \in \{i-1, i, i+1\}} D(w_k, w_j) \tag{4}$$

[7] If $k \notin [1, n]$, exclude it from the min.

This can be viewed as sacrificing precision for recall: a decision to give tokens a better chance of being able to attend to truly local tokens (given the imperfection of parser outputs), though at the cost of sometimes allowing attention on tokens that truly are not local.

Now, define a mask matrix $\mathbf{M}$ that will mask a token *iff* a token j has windowed syntactic distance over a certain threshold δ relative to token i:

$$m_{ij} = \begin{cases} 0 & \text{if } D'(w_i, w_j) \leq \delta \\ -\infty & \text{otherwise} \end{cases} \tag{5}$$

We can now define syntax-aware local attention by modifying Equation 1 so that $\mathbf{M}$ is added to the inner term in order to force an attention score of 0 for masked tokens:

$$\mathbf{A}^\ell = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}\right) \tag{6}$$

Syntax-aware local attention (SLA) is used alongside the normal, "global" self-attention. To combine the two after they have been computed, introduce a gated unit for each Transformer block with new parameters $\mathbf{W}_g$ and b_g to compute g_i for each word w_i using the word's hidden representation $\mathbf{h}_i$, where σ is the sigmoid function:

$$g_i = \sigma(\mathbf{W}_g \mathbf{h}_i + b_g) \tag{7}$$

Now, use g_i to interpolate both the normal attention distribution $\mathbf{a}_i$ and the local attention distribution $\mathbf{a}_i^\ell$ at each position i in the sequence to yield the final attention distribution $\hat{\mathbf{A}}$ and final attention head output $\hat{\mathbf{O}}$:

$$\hat{\mathbf{A}} = \bigoplus_{i=1}^{n} g_i \mathbf{a}_i + (1 - g_i)\mathbf{a}_i^\ell \tag{8}$$

$$\hat{\mathbf{O}} = \hat{\mathbf{A}}\mathbf{V} \tag{9}$$

In the original work, the SLA method is evaluated on various benchmarks on English and consistently achieves measurable improvements in model quality. Parses are obtained using Stanza (Qi et al., 2020), which for English are of quite high quality (labeled attachment score is in the mid-80s for English datasets). We refer readers to the original publication for further details. See Figure 1 for an overview.

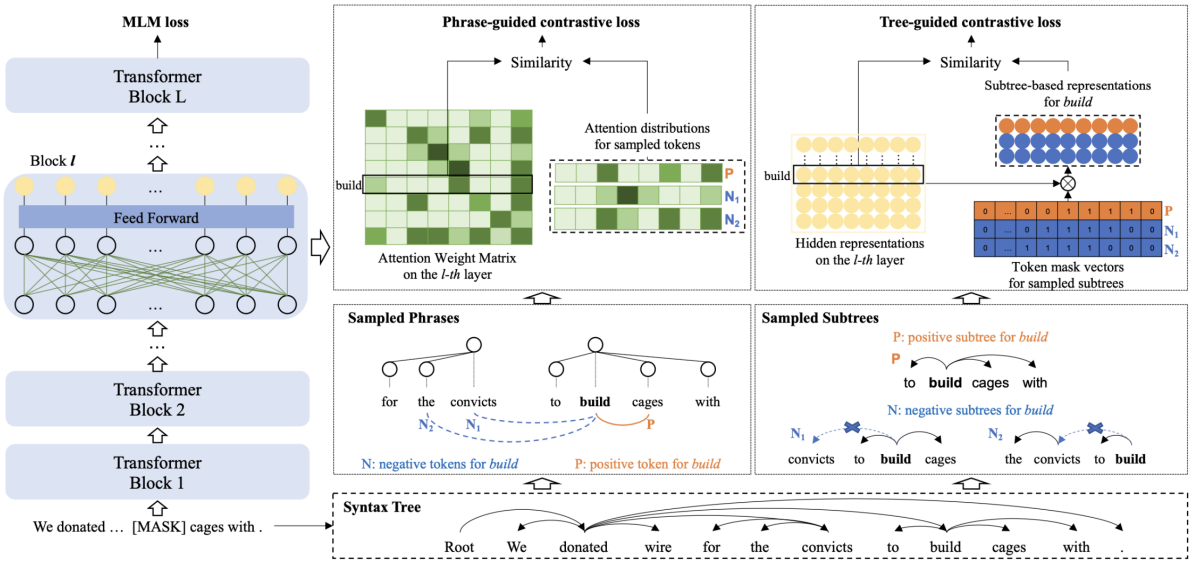

Figure 2: Figure 1 from Zhang et al. (2022b). $\boldsymbol{P}$ and $\boldsymbol{N}_i$ represent the positive sample and the ith negative sample, respectively. The phrase-based contrastive loss on the left is intended to make the representations of syntactic siblings more similar, and the tree-based contrastive loss on the right is intended to make the representations of syntactic children and parents more similar.

A.2 SynCLM

Zhang et al. (2022b) present the Syntax-guided Contrastive Language Model (SynCLM), a BERT-like TLM that characteristically uses two novel contrastive loss functions and also uses SLA (cf. appendix A.1). Intuitively, a contrastive learning objective requires each instance to have one or more *positive* and *negative* "samples", and attempts to maximize the instance's similarity to positive samples and minimize its similarity to negative samples (Zhang et al., 2022a). SynCLM uses a popular loss function for this, InfoNCE (van den Oord et al., 2018):

$$\mathrm{L} = -\log \frac{\exp\left(\frac{\mathrm{sim}(q,q^+)}{\tau}\right)}{\exp\left(\frac{\mathrm{sim}(q,q^+)}{\tau}\right) + \sum_{i=0}^{K} \exp\left(\frac{\mathrm{sim}(q,q_i^-)}{\tau}\right)} \tag{10}$$

q, q^+, and q^- are the representations of the instance, a positive sample, and a negative sample, respectively, and $\tau \in (0,1)$ is a temperature hyperparameter, set to 0.1 for SynCLM. sim is a similarity function, such as cosine similarity or KL-divergence. The loss terms obtained from this equation are simply added to the loss obtained from masked language modeling. We review only the contrastive objective functions here, and refer readers to Figure 2 and the original paper for further details.

The two SynCLM contrastive learning objectives are distinguished by how they formulate sim. The first, "phrase-guided" objective aims to make attention distributions more similar for words in the same phrase. Given a token t, sample a positive token t^+ such that t and t^+ have a lowest common ancestor t_a whose corresponding subtree (the "phrase") is no more than 2 in height. Now sample k negative tokens $t_1^-, \ldots, t_k^-$ outside the phrase, i.e. who do not have t_a as an ancestor. Define $\mathrm{sim}_{\mathrm{phrase}}$ using Jensen–Shannon Divergence (Endres and Schindelin, 2003), a similarity metric for probability distributions:

$$\mathrm{sim}_{\mathrm{phrase}} = -\mathrm{JSD}(\mathbf{a} \parallel \mathbf{a}') \tag{11}$$

Here, $\mathbf{a}$ is the attention distribution for t, and $\mathbf{a}'$ is the attention distribution for either a positive or a negative sample. This equation is used to calculate similarities for a given attention head and layer—in SynCLM's implementation, only the last layer is used, and $\mathrm{sim}_{\mathrm{phrase}}$ is averaged across all attention heads in the last layer before being used with Equation 10 for the final loss computation.

The "tree-guided" objective proceeds similarly. A token t_i is sampled which forms the root of the positive tree, T^+. Next, up to three tokens $t_1^-, \ldots, t_k^-$ are sampled such that each t_i^- is not in T^+ but is adjacent to a token in T^+. A new negative subtree T_i^- is formed for each t_i^- such that a random non-root token in T^+ has been removed from T^+ along with its children, and the subtree rooted at t_i^- has

taken its place.

We may now define tree similarity as follows, where T is a positive or a negative subtree and $\mathbf{z}_a$ is the hidden representation of token a:

$$\begin{aligned} \text{sim}_{\text{tree}} &= \text{cossim}\left(\mathbf{z}_i, \textstyle\sum_{t_j \in T_{\text{child}}} e_{ij}\mathbf{z}_j\right) \\ \text{where} \quad T_{\text{child}} &= T \setminus \{t_i\} \\ e_{ij} &= \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j\right)}{\sum_{t_k \in T_{\text{child}}} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_k\right)} \end{aligned} \tag{12}$$

Informally, we are taking the dot product of the root of the subtree with all other tokens in the subtree, softmaxing this dot product, using it to produce a weighted sum of all hidden representations of tokens in the subtree, and taking the cosine similarity between this weighted sum and the root of the subtree. The closer these tokens' representations are in the hidden space, the higher this similarity measure will be. Again, SynCLM uses only the last TLM layer for this objective, and this similarity measure is used with Equation 10. Note that in a preprocessing step, parses are modified so that subword tokens are syntactic children of the head token of the word they belong to.[8]

B English Experiments

Parsing Parsing results are given in Table 5. First note that as before, there is little difference in model quality across all the SynCLM conditions, providing more evidence that the SynCLM losses are not helpful for UD parsing. Next, as could be expected, the model trained with 100M tokens that is half the size of BERT-base performs best. What is surprising, however, is that of the remaining 3 models, the model with the standard parser performs best. Since all three of these variants are alike in model hyperparameters, this must be explainable in terms of properties of the three datasets. It could be that AMALGUM's very deliberate construction from eight genres in equal proportion could have led to serendipitously good performance on the parsing task, but it is impossible to know without further experimentation.

At any rate, whatever the differences in these three variants might be caused by that lies in the data, we still have a firm answer for our most important question: for English UD parsing, SynCLM and SLA methods appear not to be sensitive to data quantity or parse quality. The latter might be surprising, but it is worth remembering that the authors of these methods designed their algorithms in ways that may mitigate the deleterious effects of lower-quality syntactic parses. SLA uses windowed syntactic distance (cf. Equation 4 in Appendix A) for the express purpose of accommodating bad parses, and the SynCLM losses place low limits on tree height, which would help in accommodating bad parses since edges at the local, phrase level are often more reliable than edges at the clausal or inter-clausal level.

NER Results on NER are given in Table 6. Surprisingly, the same half-sized BERT model that was trained on 100M tokens and did best in the parsing evaluation does very poorly in the NER task. We suspect that this may be due to the fact that larger models can show greater instability in fine-tuning setups (Rogers et al., 2020). As with parsing, we see that the -NP model performs best among the MicroBERT-sized models, which we ascribe to differences in properties of the pretraining datasets.

What is most interesting in the NER results is that for the two low-data conditions, -NP and -HQP, we see about a 1% gain in the -MPT condition relative to the MLM-only baseline. This gain is not seen in the higher-data conditions, where none of the SynCLM combinations lead to a better model except for µB-MPT-BD, with a gain of 0.45% F1. Complicating this picture, though, is that in the low-data settings, the -MP and -MT variants often underperform relative to the baseline. Still, these results seem to indicate at least that the SynCLM loss functions may be less effective in improving model quality as quantity of pretraining data increases. We can see that this holds both for the half-sized BERT model as well as the MicroBERT-sized model, indicating that model size does not matter.

Discussion Returning to RQ3, these results indicate that SynCLM and SLA are not especially sensitive to parse quality, and are also not sensitive to model size, but are sensitive to quantity of pretraining data. As discussed above, the insensitivity to parse quality is understandable, as the dimensions in which a parse may be bad are less relevant for these methods because of the way they use the parse trees. The sensitivity to pretraining data quantity is intuitive if we consider these two methods as

[8] We have elided various implementation details here, such as hyperparameters which control how many sample sets to obtain per input sequence, or maximum token count for a subtree. Please refer to our code or Zhang et al. (2022b)'s code for these details.

Model	-NP	-HQP	-BD	-BD-BM	Avg.
μB-M	86.79	85.60	85.81	87.83	86.51
μB-MP	86.89	85.36	85.91	87.73	86.47
μB-MT	86.51	85.83	85.93	87.10	86.34
μB-MPT	86.57	85.39	85.83	86.99	86.19
μB-MPT-SLA	86.61	85.42	85.62	86.53	86.05
Avg.	86.67	85.52	85.82	87.23	

Table 5: Labeled attachment score (LAS) for English.

Model	-NP	-HQP	-BD	-BD-BM	Avg.
μB-M	60.07	58.79	57.18	51.15	56.80
μB-MP	59.99	55.29	54.46	50.96	55.18
μB-MT	56.92	55.65	57.58	49.52	54.92
μB-MPT	61.54	59.32	55.63	49.98	56.62
μB-MPT-SLA	61.49	56.05	59.51	43.90	55.24
Avg.	60.00	57.02	56.87	49.10	

Table 6: Span-based F1 score by language and model combination for NER evaluation.

sources of inductive bias: an inductive bias ought to be pushing a model towards learning something that they would have learned if there were more training data available, and so we should expect that if we consider a modification to be an inductive bias, its influence should wane as the quantity of data increases.

In sum, these findings support our conclusion that SynCLM and SLA are at least in some respects well-suited to aid the pretraining of TLMs in low-resource settings, as we have found that even when parse quality is worse than ideal, SynCLM and SLA still perform about as well as when they have the highest quality parses.

C Limitations

The goal of this paper is to make progress towards more effective TLMs for low-resource languages using syntactic inductive bias. We believe we have presented compelling evidence that two approaches to this problem seem not to be very effective for low-resource languages. But it is important to point out that we have tested the methods on only 5 languages. We believe that this forms an informative picture for low-resource languages in general because these languages are quite different from one another along typological and phylogenetic dimensions, but in principle, it is conceivable that other low-resource languages could exhibit behaviors that are very different from the ones we have seen in this paper. Moreover, we have had to re-implement the methods at the center of this work, and while we have done everything we can to ascertain that these re-implementations have been faithful and without error, tensor programming is error-prone work, and it is not impossible that we may have introduced a bug somewhere which critically affected the experimental results in this work.

D Other PrOnto Results

	Proper Noun Subject			
Model	An. Grk.	Coptic	Uyghur	Wolof
μB-M*	76.32	78.76	81.30	90.36
μB-MX*	81.11	80.78	78.45	90.36
MBERT	81.42	75.50	80.35	91.65
μB-M	79.88	79.22	80.35	80.15
μB-MP	79.72	79.38	80.82	77.97
μB-MT	79.57	75.66	81.14	77.72
μB-MPT	76.32	79.53	77.02	77.72
μB-MX	81.27	81.40	80.67	81.84
μB-MXP	78.79	78.91	79.71	79.42
μB-MXT	76.32	80.47	73.53	77.72
μB-MXPT	80.80	80.16	79.40	77.72

Table 7: Accuracy by language and model combination for Proper Noun Subject in PrOnto. Scores for MBERT, μB-M*, and μB-MX* are taken from Gessler (2023)—the asterisk indicates that the latter two models are not our implementation but the one provided in Gessler and Zeldes (2022), which is reported in Gessler (2023).

	Sentence Mood			
Model	An. Grk.	Coptic	Uyghur	Wolof
µB-M*	90.18	89.75	89.96	90.36
µB-MX*	91.56	89.75	90.10	90.36
MBERT	91.70	91.55	91.23	91.65
µB-M	91.98	91.69	91.51	90.36
µB-MP	90.73	91.97	91.23	89.72
µB-MT	90.59	90.30	89.25	90.36
µB-MPT	90.73	90.30	89.96	90.36
µB-MX	90.59	92.24	91.80	90.58
µB-MXP	91.56	91.97	90.81	90.58
µB-MXT	91.42	90.03	89.96	90.36
µB-MXPT	90.73	90.03	89.96	90.36

Table 8: Accuracy by language and model combination for Sentence Mood in PrOnto. Scores for MBERT, µB-M*, and µB-MX* are taken from Gessler (2023)—the asterisk indicates that the latter two models are not our implementation but the one provided in Gessler and Zeldes (2022), which is reported in Gessler (2023).

	Same Argument Count			
Model	An. Grk.	Coptic	Uyghur	Wolof
µB-M*	61.80	62.70	61.78	61.05
µB-MX*	61.71	61.58	62.12	63.46
MBERT	50.87	51.24	50.78	54.46
µB-M	59.72	56.94	59.23	56.65
µB-MP	58.57	57.61	59.99	58.38
µB-MT	58.44	57.26	59.43	56.10
µB-MPT	53.13	56.01	59.56	56.32
µB-MX	57.06	55.92	60.10	56.05
µB-MXP	58.60	56.18	59.87	56.21
µB-MXT	58.03	56.47	59.67	56.69
µB-MXPT	58.36	58.01	57.88	57.54

Table 9: Accuracy by language and model combination for Same Argument Count in PrOnto. Scores for MBERT, µB-M*, and µB-MX* are taken from Gessler (2023)—the asterisk indicates that the latter two models are not our implementation but the one provided in Gessler and Zeldes (2022), which is reported in Gessler (2023).