Scaling Laws for BERT in Low-Resource Settings

Gorka Urbizu[1] **Iñaki San Vicente**[1] **Xabier Saralegi**[1]

Rodrigo Agerri[2] **Aitor Soroa**[2]

[1]**Orai NLP Technologies**
[g.urbizu|i.sanvicente|x.saralegi]@orai.eus

[2]**HiTZ Center - Ixa, University of the Basque Country UPV/EHU**
[rodrigo.agerri|a.soroa]@ehu.eus

Abstract

Large language models are very resource intensive, both financially and environmentally, and require an amount of training data which is simply unobtainable for the majority of NLP practitioners. Previous work has researched the scaling laws of such models, but optimal ratios of model parameters, dataset size, and computation costs focused on the large scale. In contrast, we analyze the effect those variables have on the performance of language models in constrained settings, by building three lightweight BERT models (16M/51M/124M parameters) trained over a set of small corpora (5M/25M/125M words). We experiment on four languages of different linguistic characteristics (Basque, Spanish, Swahili and Finnish), and evaluate the models on MLM and several NLU tasks. We conclude that the power laws for parameters, data and compute for low-resource settings differ from the optimal scaling laws previously inferred, and data requirements should be higher. Our insights are consistent across all the languages we study, as well as across the MLM and downstream tasks. Furthermore, we experimentally establish when the cost of using a Transformer-based approach is worth taking, instead of favouring other computationally lighter solutions.

1 Introduction

Pre-trained neural language models based on the Transformer architecture have shown impressive results on many NLP tasks to the point that their use has become standard practice. The capabilities of these models improve as the complexity (in terms of parameters) of their architecture (Wei et al., 2022a) and the size of the corpora on which the pre-training is performed increase (Zhang et al., 2021). For this reason, there is now a tendency to build ever-larger models trained on ever-growing corpora. This trend has resulted in a never-ending increase of the computational requirements to perform model pre-training, but also for the subsequent fine-tuning and inference processes at production time. Moreover, building very large models require huge training corpora, which is only available for a handful of rich-resource languages.

Kaplan et al. (2020) and Hoffmann et al. (2022) propose power-law formulas that relate model size, corpora size and computation power, and help find the optimal settings in advance given a fixed budget. However, their analysis is focused on autoregressive models of relatively big sizes, that require large corpora to train. In this paper, we analyze whether the conclusions drawn in these works also apply to discriminative (encoder-only) models in low-resource settings, where both the data size and budget are constrained. We analyze the performance of several combinations of model and data sizes using a simulated low-resource scenario in four linguistically diverse languages from different families (Basque, Spanish, Swahili and Finnish).

Our study reveals that the data size and model size power law values provided by Kaplan et al. (2020) and Hoffmann et al. (2022) are not optimal in these scenarios. Instead, our experiments show that data size should be relatively bigger than what those scaling laws estimated when training small models (16-124M parameters) for optimal performance. Furthermore, given a fixed computational budget, it is better to train big models instead of

Findings of the Association for Computational Linguistics: ACL 2023, pages 7771–7789
July 9-14, 2023

computing more model updates in smaller models.

Additionally, we establish the minimally required combinations of compute, model and dataset sizes of Transformer-based approaches that outperform other lighter neural baselines, taking CO_2 emissions into consideration.

2 Related Work

Since the emergence of the attention-based Transformer (Vaswani et al., 2017) architecture and the masking pre-training strategies introduced since BERT (Devlin et al., 2019), different pre-training strategies have been published. But aside from the improvements to the architecture or training procedures, the qualitative improvement in results is mainly the result of increasing the model size, alongside the amount of text corpora used to train them: Chinchilla (70B parameters) (Hoffmann et al., 2022), LaMDA (137B) (Thoppilan et al., 2022), GPT-3 (175B) (Brown et al., 2020), Gopher (280B) (Rae et al., 2021) and PaLM (540B) (Chowdhery et al., 2022). This fast-growing increase in model sizes and data is proven to surface new abilities in larger models, but not present in smaller models (Wei et al., 2022a).

The relationship between the size of the pre-training corpus and the performance of the language model in NLU tasks has been addressed in the literature before. The performance improves when the amount of data is increased (Zhang et al., 2021; Hu et al., 2020; Micheli et al., 2020; Raffel et al., 2020), although, at a certain point, the increase in performance slows down when the model size is kept fixed (Inoue et al., 2021; Martin et al., 2020; Micheli et al., 2020; Raffel et al., 2020; Liu et al., 2021). Furthermore, it is more convenient to improve the diversity of training datasets, than to add more text from the same domains (Inoue et al., 2021; Martin et al., 2020; Liu et al., 2021).

The correlation between model size and model performance on NLU tasks has also been analyzed. The performance of the model improves when scaling the model size (and FLOPs) (Turc et al., 2019; Raffel et al., 2020; Xia et al., 2022; Clark et al., 2022). However, all these works used very large pre-training datasets. They do not analyze if the increase in performance slows down bottlenecked by the pre-training dataset size and thus, the conclusion of scaling being always beneficial cannot be extended to low-data scenarios.

The works of Kaplan et al. (2020) and Hoffmann et al. (2022), whose aim is aligned with this work, empirically study the optimal ratios of the training tokens, model parameters, and computation to train dense language models and infer scaling laws.

Kaplan et al. (2020) train models of a size ranging from 768 to 1.5 billion parameters with datasets ranging from 22 million to 23 billion tokens and conclude that LM performance improves smoothly as we increase the model size, dataset size, and amount of computation. They show that all three factors must be scaled up in tandem, to avoid bottleneck issues. Furthermore, they note that larger models are more sample-efficient, and that convergence is inefficient, suggesting that it's better to under-train a bigger model than converge a smaller one on the same computing budget.

Hoffmann et al. (2022) find the optimal model size and the number of training tokens for given a fixed FLOPs budget. For this purpose, they draw their own scaling laws, based on the losses of over 400 models, ranging from 70M to 16B parameters, and trained on 5B to 400B tokens. They state that model size and the number of training tokens should scale equally, based on three alternative approaches, while Kaplan et al. (2020) extrapolates that every time the model size is increased by 8, the data only needs to be increased by 5. Thus, after concluding that the performance of most of the current large language models is bottlenecked by the undersized corpora, they train Chinchilla.

However, the scaling laws of Kaplan et al. (2020) and Hoffmann et al. (2022) are not useful for the low-resource settings we want to focus on. According to Kaplan et al. (2020) we need very small training corpora (e.g. 744K tokens for a BERT$_{BASE}$, which is clearly not enough or optimal).

Hoffmann et al. (2022) infers significantly bigger training corpora: e.g. 3M tokens for a BERT$_{MINI}$ or 86M tokens for a BERT$_{BASE}$. Current models for low-resource languages are trained with corpora around that range (Joshi et al., 2020): 161M for Irish (Barry et al., 2021), 130M for Luxembourgish (Lothritz et al., 2022), 45M for Galician (Vilares et al., 2021), 16M for Swahili (Martin et al., 2022b) and 4.4M for Quechua (Zevallos et al., 2022). However, increasing pre-training data several orders of magnitude has been proved beneficial for base size models (Liu et al., 2019).

Finally, those optimal scaling laws have been deduced from models trained over one epoch, while in low/medium-resource settings models are often

trained over several epochs (Martin et al., 2020; Lothritz et al., 2022; Zevallos et al., 2022).

Nonetheless, for certain NLU tasks (e.g. NeQA and Quote Repetition) scaling language models is detrimental (Perez and McKenzie, 2022), creating inverse scaling laws. However, Wei et al. (2022b) implies U-shaped scaling laws where even larger models might be able to solve those tasks that comprise a true and a distractor task.

3 Experimental Setup

We aim to find the optimal combination of model-size, dataset-size and computing in low-resource environments and assess whether they follow the scaling laws established in the literature. In addition, we seek to find the minimum requirements to overcome computationally lighter neural baselines. Therefore, we carry out experiments for 3 corpus sizes and 3 model sizes, in 4 languages, training a total of 36 different models.

3.1 Language Selection

We conduct the experiments in four languages from different language families, selected among those that have enough monolingual data to train LMs, as well as enough available evaluation datasets for NLU tasks. Hence, the low-resource setting has been simulated in some cases. Among other languages that fulfil those criteria, we opted for Basque (eu), Spanish (es), Swahili (sw) and Finnish (fi). In addition to being part of disjoint language families, these languages are linguistically diverse with different complexities across morphology, syntax, verb system and vocabulary (Coloma, 2015) (see Appendix A).

3.2 Corpora

For each language, we created three corpora comprising 125M, 25M and 5M words, respectively. We limited the number of corpora sizes to three in order to control the number of experiments, and thus the computational resources needed.

Preliminary experiments showed a big fall in the results when reducing pre-training data to just 1M words, in consistency with Zhang et al. (2021). Since obtaining corpora of about 5M words is achievable by most languages that have annotated datasets (Joshi et al., 2020), we set the lower bound at 5M words. Zhang et al. (2021) shows that 10M to 100M words of pretraining data are enough for a language model to acquire the linguistic capacities of syntax and semantics. Thus, we set the other two corpora sizes at 25M and 125M words, keeping a constant increase rate among them.

Regarding the nature of the texts, corpora for Basque and Spanish are a mix of 75% news and 25% text from Wikipedia. We selected the newspaper Berria[1] for Basque, and El Pais[2] for Spanish. Corpora for Swahili and Finnish were built by randomly selecting documents (longer than 10 sentences) from the web-crawled *cc100* corpus (Conneau et al., 2020; Wenzek et al., 2020).

	L	HH	INT	H	NEP	Param.
BERT_{124M}	12	768	3072	12	86M	124M
BERT_{51M}	8	512	2048	8	25M	51M
BERT_{16M}	4	256	1024	4	3M	16M

Table 1: Model sizes in our experiments. L: layers. HH: hidden dimensions. INT: intermediate layer dimensions. H: attention heads. NEP: non-embedding parameters.

3.3 Models

In a similar fashion to (Turc et al., 2019), we employ three different variants of the BERT model, dubbed BERT_{124M}[3], BERT_{51M} and BERT_{16M}. These models have 12, 8 and 4 layers respectively, also shrinking other parameters proportionally (hidden dimension, number of attention heads, etc.), since model shape does not affect performance significantly (Kaplan et al., 2020). Table 1 shows a detailed view of the parameters in each model. We also increased the vocabulary sizes from the original 30K subword tokens to 50K because it is beneficial for agglutinative languages (Agerri et al., 2020). We trained each model up to 500K steps with a batch of 256 and a sequence length of 512. For more pre-training details see Appendix B.

4 Evaluation Settings

We evaluate all models intrinsically and extrinsically. For the intrinsic evaluation, we tested the models on masked language modeling; for the extrinsic evaluation, we selected four NLU downstream tasks with available datasets in all the selected languages: Name Entity Recognition and Classification (NERC), Topic Classification (Topic), Sentiment Analysis (SA) and Question-answering NLI (QNLI). Our selection of tasks in-

[1] https://www.berria.eus

[2] https://elpais.com

[3] This corresponds to BERT_{base} in Devlin et al. (2019)

	EU			ES			SW			FI			
Task	train	dev	test	train	dev	test	train	dev	test	train	dev	test	M.
NERC	52K	13K	36K	265K	53K	52K	175K	25K	51K	180K	14K	46K	F1
Topic	9K	2K	2K	9K	1K	4K	10K	3K	7K	10K	10K	10K	F1
SA	6K	1K	1K	5K	2K	1K	6K	782	1K	4K	633	1K	F1
QNLI	2K	230	238	30K	4K	4K	4K	624	1K	7K	1K	1K	acc
MLM			1M			1M			1M			1M	acc

Table 2: Datasets used in the evaluation. The size for NERC and MLM are not reported in examples but in tokens. F1 refers to the micro-average F1-score, while acc refers to accuracy.

cludes one sequence labeling task and three sequence classification tasks including sentiment analysis and QNLI, which are tasks that require a deeper NLU than the shallow linguistic tasks of NERC and Topic Classification (Zhang et al., 2021). Table 2 shows the details of each dataset.

4.1 MLM

Masked Language Modeling (MLM) is one of the default pre-training objective functions of BERT. We report both the loss and accuracy of MLM.

For this purpose, we created test datasets, from news sources not used for pre-training the models. For Basque we gathered texts from Argia[4] news magazine. For Spanish, we opted for texts from the newspaper El Mundo [5]. For Swahili, as a data source not used in the pre-training, we randomly selected a sub-corpus from the pre-train data for SwahBERT model (Martin et al., 2022a), which is mostly made up of news (%80). For Finnish we opted for a subset of *cc100* not used in the pre-training, due to the lack of document-level news corpora available with an open license.

4.2 NERC

Named Entity Recognition and Classification (NERC) is a token classification task. For Basque, we used the *in-domain NERC* dataset from the BasqueGLUE benchmark (Urbizu et al., 2022). For Spanish, we opted for the *Conll2002* dataset (Sang, 2002). For Swahili, we selected *Masakhaner* (Adelani et al., 2021). And lastly, for Finnish, we used *FiNER* (Ruokolainen et al., 2019). Each dataset has 4 categories, and we use the F-score as the performance metric.

4.3 Topic Classification

Topic classification is a sequence classification multi-class task. For Basque, we chose the *BHTCv2* dataset including 12 thematic classes (Urbizu et al., 2022). The Spanish counterpart is *MLdoc* (Schwenk and Li, 2018) which has 4 classes. For Swahili, we employed *Swahili: News Classification Dataset* (David, 2020) which has 4 thematic classes. Since a development dataset split was missing, we randomly selected the 20% of the training split to create it. Furthermore, since the fine-tuning dataset is bigger than the smallest of the pre-training dataset (5M), we downsampled this training to 10K examples. And for Finnish, we selected the 10% version of the *Yle corpus*[6], which contains 10 thematic classes. Performance is measured with the F-score score.

4.4 SA

Sentiment Analysis (SA) is a sequence classification task. For Basque, we employed the dataset *BEC2016eu* (Urbizu et al., 2022), which has positive, negative and neutral classes. *InterTass2020* (Cumbreras et al., 2016) is the Spanish dataset selected for SA, which also has positive, negative and neutral classes. For Swahili, we utilized the dataset presented by Martin et al. (2022b). The dataset was mapped to polarity annotation following guidelines from the article: joy ([1]) = positive, disgust ([4]) = negative, neutral ([0]) and surprise ([5]) = neutral. Only examples with a single label were mapped. Original train/dev/test splits were maintained. And lastly, for Finnish, we chose *Finnish sentiment*[7] which only contains positive and negative labels. We use F-score as the performance metric.

4.5 QNLI

Question-answering NLI (QNLI) is a sequence classification task. For Basque, we employed $QNLI_{eu}$ (Urbizu et al., 2022). And for Spanish, Swahili and Finnish, we adapted already available

[4] www.argia.eus
[5] www.elmundo.es
[6] www.github.com/spyysalo/yle-corpus
[7] www.huggingface.co/datasets/sepidmnorozy/Finnish_sentiment

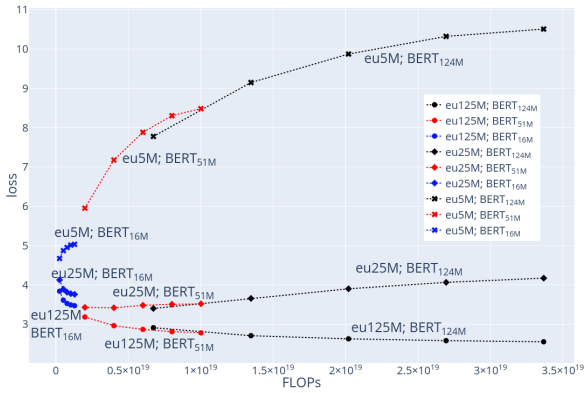

Figure 1: MLM loss and FLOPs for models for Basque.

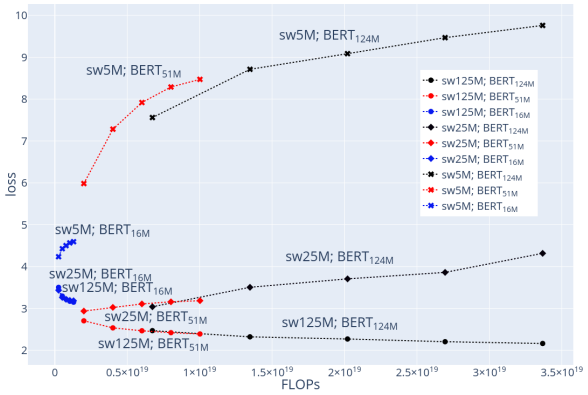

Figure 3: MLM loss and FLOPs for models for Swahili.

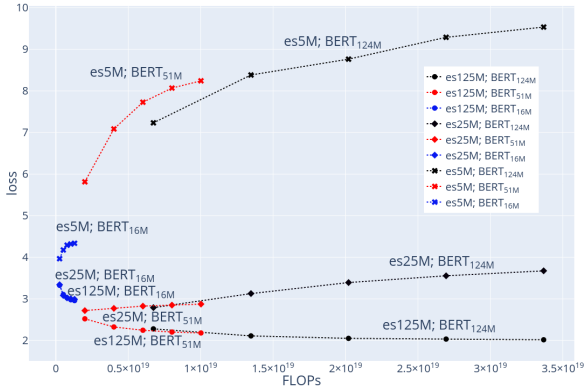

Figure 2: MLM loss and FLOPs for models for Spanish.

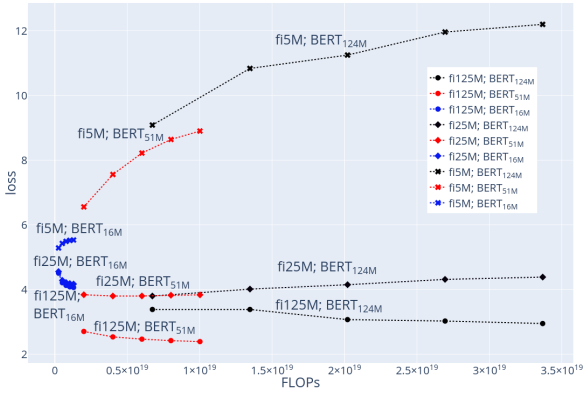

Figure 4: MLM loss and FLOPs for models for Finnish.

conversational Question Answering (QA) datasets, into a sequence-pair binary classification task following the design of QNLI for English (Wang et al., 2019). The QA dataset selected were *SQAD*$_{es}$ (Carrino et al., 2020), *Tydiqa*$_{sw}$ and *Tydiqa*$_{fi}$ (Clark et al., 2020). Each QNLI dataset has a Question-Sequence pair for entailment[8]. *Tydiqa* only provides splits for train and development. Thus, we used that development split as our test split, and randomly select some examples from the training set to create our development set. We follow the English QNLI design and use accuracy as the evaluation metric.

4.6 Systems and Baselines

For the extrinsic evaluation, we fine-tuned each of the 36 BERT models making use of the Transformers library (Wolf et al., 2020), with a lr of $3e^{-5}$, an effective batch size of 32 and training up to 10 epochs[9], which are considered default values (Devlin et al., 2019; Mosbach et al., 2020). For each task and language, we report an average of 5 runs.

In order to compare the performance of our models to other lighter approaches, we implement a competitive neural baseline based on contextual embeddings using Flair (Akbik et al., 2019). For sequence labeling tasks, embeddings are passed into a BiLSTM-CRF system based on the architecture proposed by (Huang et al., 2015). For text classification tasks, the computed Flair embeddings are fed into a BILSTM to produce a document-level embedding which is then used in a linear layer to make the class prediction.

We pre-train our own contextual Flair embeddings using the 125M corpora for each language with the following hyperparameters: Hidden size of 2048, sequence length of 250, a mini-batch size of 100 and 10 epochs. The rest of the training parameters are left in their default setting[10].

In addition to LSTM-based neural baselines, we also include in the comparison a multilingual BERT

[8]In the case of Basque and Spanish, those sequences are a single sentence, while the sequences for Finnish and Swahili are short paragraphs, because the QA datasets were annotated following different methodologies.

[9]Selecting the best-performing epoch on the development set.

[10]Each model took 80h to train on an Nvidia TitanV GPU.

model, namely mBERT$_{base}$ (Devlin et al., 2019). We assume that this kind of multilingual language models will always be available and that they can somehow be an alternative in some low-resource settings, namely when limited resources refer to the computational capacity for pre-training and to the availability of enough text in the target language. In that line, we perform this comparison only for the Basque and Swahili languages which approximately include a training corpus size in mBERT no larger than those of the corpora used in our experiments (roughly 35M for eu and 11M for sw).

5 Results

5.1 Down-Scaling Laws

Figures 1-4[11] show the relation between the FLOPs and the MLM loss in the development dataset for the models in the four languages. Different colours stand for different model sizes (16M/51M/125M), and different symbols represent the pre-training data (5M/25M/125M). Each line is formed by 5 checkpoints (every 100K steps).

We can appreciate how the lowest loss is achieved by the biggest model with the most pre-training data, as expected from scaling laws, which dictate an improvement in performance when model-size, dataset-size and compute budget are increased simultaneously. Furthermore, the plot shows that increasing pre-training data is more beneficial than increasing the model size or amount of compute in this low-resource scenario; the gap between the loss obtained from increasing pre-training data is much bigger than the improvements obtained when increasing model-size or training steps.

The figures also show that models trained with small datasets yield larger MLM losses in development with further training ($\times$ curves), which we attribute to overfitting, as the training MLM loss does shrink as training advances. Big models with medium datasets (red and black $\diamond$ curves) also show the same tendency, although to a lesser extent. According to the figures, unless we use a dataset of at least 125M in the case of BERT$_{51M}$ and BERT$_{124M}$, or a dataset of at least 25M in the case of the smallest BERT$_{16}$, we should consider applying early stopping to our models to avoid overfitting, which corresponds with the first checkpoint of 100K steps we plotted in most cases.

Nevertheless, over-fitting issues during pre-training, do not have a direct impact neither on MLM accuracy nor on downstream tasks (see Appendix D). Thus, for the evaluation of the models regarding MLM accuracy (analysis available in Appendix C) and NLU downstream tasks (Section 5.2), we employed the final checkpoints at 500K steps.

Regarding languages, a comparison of the four figures (Figures 1-4) shows that the correlation between MLM loss and combinations of model-size, dataset-size and FLOPs is consistent across languages. MLM accuracies are also consistent across languages (Appendix C).

Hoffmann et al. (2022) estimates that 3M, 25M and 86M tokens are optimal to train BERT$_{16M}$ BERT$_{51}$ and BERT$_{124}$ respectively, while Kaplan et al. (2020) estimates much lower values. Our results show that the amount of data needed to train an LM optimally is no less than 25M words for BERT$_{16M}$ and 125M for BERT$_{51}$ and BERT$_{124}$. We carry out an in-depth comparison of these results with additional data in Appendix H.

5.2 Evaluation on NLU tasks

This section analyses the performance of our models on the NLU tasks listed in section 4, to measure the effect of model-size and pre-training data-size once finetuned on the downstream tasks.

The results for the NERC task are shown in Table 3. As expected, there is a clear positive correlation between the evaluation metric and the model and corpora size, but corpora size has slightly more impact on the performance. The results for topic classification at Table 4 follow the same trends, albeit with smoother differences. Table 5 presents the results for SA, again repeating the trends, but with a few outliers. Lastly, for QNLI (see Table 6), we observe there is a general trend of improving results while increasing dataset and model sizes. However, many results present large standard deviations, leading to several outliers[12] that stand out from the general trend.

The models trained obtain competitive results, as shown by the results for NERC, topic classification and SA for Swahili, which are new SotA for those datasets to the best of our knowledge. Besides, some of the results obtained with the BERT$_{124M}$ and 125M words are comparable with SotA models trained over huge datasets (See appendix F).

[11] Zoomed in for BERT$_{16M}$ in the Appendix G.

[12] outliers marked in red

NERC$_{eu}$	5M	25M	125M
BERT$_{16M}$	63.90±0.5	72.23±0.6	74.12±0.3
BERT$_{51M}$	70.14±0.4	79.07±0.4	82.98±0.1
BERT$_{124M}$	73.14±0.5	79.09±0.8	**84.58**±0.2
NERC$_{es}$	5M	25M	125M
BERT$_{16M}$	76.57±0.3	81.56±0.5	81.70±0.5
BERT$_{51M}$	80.43±0.4	85.11±0.8	86.34±0.7
BERT$_{124M}$	81.75±0.4	84.99±0.8	**87.28**±0.3
NERC$_{sw}$	5M	25M	125M
BERT$_{16M}$	86.36±0.2	88.62±0.2	88.63±0.4
BERT$_{51M}$	88.74±0.2	90.68±0.2	91.63±0.1
BERT$_{124M}$	88.93±0.4	90.97±0.2	**92.09**±0.2
NERC$_{fi}$	5M	25M	125M
BERT$_{16M}$	76.82±0.3	81.48±0.4	81.83±0.3
BERT$_{51M}$	79.73±0.2	85.27±0.4	87.02±0.2
BERT$_{124M}$	80.56±0.7	85.77±0.3	**88.99**±0.2

Table 3: Results for the 9 models on NERC (F1) for Basque, Spanish, Swahili and Finnish.

Topic$_{eu}$	5M	25M	125M
BERT$_{16M}$	68.00±0.6	71.81±0.3	72.49±0.4
BERT$_{51M}$	69.98±0.6	73.16±0.6	74.87±0.4
BERT$_{124M}$	71.70±0.7	74.61±0.3	**76.06**±0.4
Topic$_{es}$	5M	25M	125M
BERT$_{16M}$	94.54±0.3	95.86±0.3	95.42±0.4
BERT$_{51M}$	94.89±0.3	95.45±0.2	95.91±0.4
BERT$_{124M}$	95.32±0.4	95.82±0.3	**96.27**±0.3
Topic$_{sw}$	5M	25M	125M
BERT$_{16M}$	91.64±0.3	91.96±0.3	92.45±0.2
BERT$_{51M}$	92.12±0.2	92.39±0.1	92.88±0.2
BERT$_{124M}$	91.95±0.4	92.69±0.1	**93.07**±0.2
Topic$_{fi}$	5M	25M	125M
BERT$_{16M}$	88.15±0.1	88.94±0.2	89.16±0.2
BERT$_{51M}$	88.53±0.3	89.40±0.3	89.61±0.3
BERT$_{124M}$	88.41±0.2	89.72±0.2	**90.14**±0.1

Table 4: Results for the 9 models on topic classification (F1) for Basque, Spanish, Swahili and Finnish.

SA$_{eu}$	5M	25M	125M
BERT$_{16M}$	67.80±0.5	68.63±1.0	67.59±0.5
BERT$_{51M}$	67.00±1.0	68.54±0.5	**69.40**±0.9
BERT$_{124M}$	67.22±0.7	68.79±0.7	68.91±0.5
SA$_{es}$	5M	25M	125M
BERT$_{16M}$	37.67±1.5	37.62±0.7	37.51±1.4
BERT$_{51M}$	36.05±2.1	37.57±0.5	39.89±0.0
BERT$_{124M}$	36.37±2.2	37.17±1.1	**43.27**±1.1
SA$_{sw}$	5M	25M	125M
BERT$_{16M}$	71.52±0.6	75.56±0.3	74.84±0.6
BERT$_{51M}$	70.49±0.7	75.39±1.4	77.07±0.0
BERT$_{124M}$	69.60±1.3	75.54±0.9	**79.04**±0.7
SA$_{fi}$	5M	25M	125M
BERT$_{16M}$	89.69±0.2	90.96±0.2	91.14±0.4
BERT$_{51M}$	89.61±0.6	91.86±0.3	92.58±0.0
BERT$_{124M}$	90.32±0.5	91.55±0.3	**94.38**±0.3

Table 5: Results for the 9 models on sentiment analysis (F1) for Basque, Spanish, Swahili and Finnish.

QNLI$_{eu}$	5M	25M	125M
BERT$_{16M}$	68.19±2.3	68.95±2.0	71.22±5.0
BERT$_{51M}$	65.06±0.8	**76.37**±2.5	74.18±1.6
BERT$_{124M}$	67.43±2.7	72.66±2.0	74.09±1.7
QNLI$_{es}$	5M	25M	125M
BERT$_{16M}$	65.01±0.6	70.89±1.4	72.72±0.7
BERT$_{51M}$	67.00±1.8	74.11±1.1	78.00±0.0
BERT$_{124M}$	67.39±0.5	73.07±1.3	**81.10**±0.7
QNLI$_{sw}$	5M	25M	125M
BERT$_{16M}$	62.80±1.1	62.45±1.2	63.42±1.0
BERT$_{51M}$	62.27±1.7	63.83±1.4	**63.87**±1.5
BERT$_{124M}$	64.08±1.1	62.68±1.2	63.34±1.4
QNLI$_{fi}$	5M	25M	125M
BERT$_{16M}$	51.49±0.9	50.96±0.7	58.89±3.7
BERT$_{51M}$	54.28±2.5	54.58±3.2	57.30±4.3
BERT$_{124M}$	54.07±2.6	**59.89**±4.5	58.56±1.1

Table 6: Results for the 9 models on QNLI (acc) for Basque, Spanish, Swahili and Finnish.

5.2.1 Evaluation vs Baseline Systems

Table 7 contains the results for the models trained with the corpora of 125M words, compared to the BiLSTM-CRF Flair baseline (trained with the same 125M corpora) and mBERT (for the languages with a comparable target-language pre-training corpus size). BERT models outperform the Flair neural baseline, but, depending on the evaluation dataset (task and language), the baseline is outperformed only by the BASE$_{124M}$ model or by all three model sizes. Furthermore, for some datasets, even the BERT$_{16M}$ models trained with the smallest corpus (5M), not included in this table, outperform the Flair baseline (trained over a corpus of 125M words). Finally, computational costs (analysed in Section 5.3) ought to be a factor to consider and decide if the gain in performance is worth the increase in computational requirements.

The boost in performance when increasing the model size is larger in downstream tasks than in the MLM intrinsic task, particularly when shifting from the smallest BERT$_{16M}$ to the intermediate

		BERT_{16M}	BERT_{51M}	BERT_{124M}	Flair	mBERT
eu	NERC	74.12±0.3	82.98±0.1	**84.58**±0.2	82.13±0.4	79.39±1.0
	Topic	72.49±0.4	74.87±0.4	**76.06**±0.4	67.89±0.3	70.57±0.5
	SA	67.59±0.5	**69.40**±0.9	68.91±0.5	68.17±0.3	67.34±0.7
	QNLI	71.22±5.0	74.18±1.6	74.09±1.7	48.66±5.2	**78.48**±1.9
es	NERC	81.70±0.5	86.34±0.7	**87.28**±0.3	87.09±0.3	—
	Topic	95.42±0.4	95.91±0.4	**96.27**±0.3	94.08±0.4	—
	SA	37.51±1.4	39.89±0.0	**43.27**±1.1	34.73±3.0	—
	QNLI	72.72±0.7	78.00±0.0	**81.10**±0.7	56.42±0.6	—
sw	NERC	88.63±0.4	91.63±0.1	**92.09**±0.2	92.04±0.1	91.17±0.1
	Topic	92.45±0.2	92.88±0.2	**93.07**±0.2	91.83±0.2	91.52±0.2
	SA	74.84±0.6	77.07±0.0	**79.04**±0.7	73.60±0.5	69.17±1.2
	QNLI	63.42±1.0	**63.87**±1.5	63.34±1.4	52.82±2.1	63.48±1.1
fi	NERC	81.83±0.3	87.02±0.2	**88.99**±0.2	84.76±0.4	—
	Topic	89.16±0.2	89.61±0.3	**90.14**±0.1	86.58±0.7	—
	SA	91.14±0.4	92.58±0.0	**94.38**±0.3	89.74±0.5	—
	QNLI	**58.89**±3.7	57.30±4.3	58.56±1.1	51.54±1.2	—

Table 7: Results for BERT and Flair models pre-trained with the biggest dataset (125M words) and mBERT (for the languages pre-trained with a comparable corpus).

BERT_{51M}. This indicates that a larger model is better suited for fine-tuning, as the number of trainable parameters is also higher.

The results and scaling trends across languages are very consistent. The results and the trends we obtained are also consistent across different tasks, with the exception of QNLI, where results are volatile[13] and have many outliers.

5.3 FLOPs and CO_2 Emissions

Table 8 shows the computational costs and CO_2 emissions for each system for training, finetuning[14] and inference. We calculated the FLOPs following the same method as Hoffmann et al. (2022). For non-transformer baselines, FLOPs were computed following (Zhang et al., 2018). CO_2 emissions were estimated with *Machine-Learning Impact calculator*[15] (Lacoste et al., 2019). The neural baseline based on Bi-LSTMs is lighter FLOP-wise, on pre-training, fine-tuning and inference time, even against the smallest BERT_{16M} model. Still, the Flair baseline has higher CO_2 emissions for fine-tuning, due to its inability to parallelize from the recurrent nature of the LSTMs.

If we revisit the results on MLM and NLU tasks (Sections 5.1 and 5.2) with computational costs in mind, we can say that if we only have a tiny corpus (5M token) available, the results obtained with a small model (BERT_{16M}) are on par with its bigger siblings at MLM, Topic, SA and QNLI, but not in NERC, where increasing the model size (up to BERT_{51M}) is needed to get competent results. In a scenario with a small dataset (25M), BERT_{16M} would only obtain comparable results at topic classification and SA, but a BERT_{51M} model obtains results as good as, or even better than BERT_{124M}. Thus, we can opt for the BERT_{51M} and use only half of the compute. However, if we are working with a pre-training dataset bigger than 125M, BERT_{124M} obtains the best results by far, indicating that it is worth investing the compute needed to train such a model.

However, here we are comparing models of different sizes, trained for the same amount of steps. What would happen if we want the best model for a fixed computational budget? We answer that in Appendix E, where we compare the BERT_{51M} and BERT_{124M}, pre-trained on a comparable amount of computation. In line with (Kaplan et al., 2020), we conclude that it is better to undertrain a BERT_{124M} than overtraining a BERT_{51M} with the same amount of computation.

6 Conclusions

We present a study of the performance of language models in constrained settings, to analyze if the same scaling laws studied for large-language mod-

[13] with an average standard deviation of 1.8

[14] Finetuning values are computed for a single run at Spanish topic classification.

[15] https://mlco2.github.io/impact#compute

	Pre-training		Fine-tuning		Inference	
Model	FLOPs	CO_2eq	FLOPs	CO_2eq	FLOPs	CO_2eq
BERT_{124M}	$4.9e{+}19$	98 kg	$3.4e{+}16$	47 g	$1.3e{+}11$	0.18 mg
BERT_{51M}	$2.0e{+}19$	41 kg	$1.4e{+}16$	23 g	$5.3e{+}10$	0.07 mg
BERT_{16M}	$6.3e{+}18$	13 kg	$4.4e{+}15$	11 g	$1.6e{+}10$	0.02 mg
Flair	$1.4e{+}17$	4 kg	$5.3e{+}14$	334 g	$5.3e{+}09$	0.01 mg

Table 8: Computational costs in FLOPs for pre-training, fine-tuning and inference and their estimated CO_2 emissions.

els apply to low-resource scenarios. We find out that the estimated values for optimal balance of model size and corpora size do not hold in these scenarios, and that pre-training tokens should be higher than the amount of model parameters.

From our experiments, we conclude that it is preferable to train big models on as much data as possible rather than using the computational power to further train smaller models. We see a clear trend where bigger models tend to quickly overfit when pre-trained for many epochs with small corpora. Still, even when they overfit in the pre-training stage, bigger models consistently outperform smaller models in downstream applications which require fine-tuning.

The experimental results are consistent among languages. Additionally, we empirically establish when the computational cost of using a Transformer-based approach is worth taking.

All the pre-training corpora, models and datasets created in this work are publicly available[16].

Limitations

First of all, our study is limited to languages that use the Latin script. Still, the 4 languages are from different language families and are typologically diverse.

Secondly, the low-resource scenario is simulated. As mentioned in 3, in order to carry out the experiments the languages involved were required to have enough monolingual data to train LMs, as well as available evaluation datasets for NLU tasks.

The source of the pre-training corpora for Swahili and Finnish (*cc100*) is not completely comparable with the corpora used for Basque and Spanish (75% news, 25% Wikipedia), due to the unavailability of a large curated corpus for Swahili, and the lack of big news corpora for Finnish with an open license that allowed us to share freely the pre-training data.

Our study is limited to 3 language model sizes and 3 pre-training corpora sizes. Including other model sizes like a BERT-Large or a model between 51M and 16M (where there is a big gap in results), and adding more pre-training corpora sizes (let's say 625M and 1M words) were out of the scope of this work.

In addition, we use the default hyperparameters that are commonly used for BERT-base (BERT_{124M}) for the pre-training and fine-tuning of the BERT_{51M} and BERT_{16M} models without any hyperparameter tuning.

Acknowledgements

This work has been partially funded by the Basque Government (ICL4LANG project, grant no. KK-2023/00094). It has also received funding from the following MCIN/AEI/10.13039/501100011033 projects: (i) DeepKnowledge (PID2021-127777OB-C21) and ERDF A way of making Europe and, (ii) DeepR3 (TED2021-130295B-C31) and European Union NextGeneration EU/PRTR. Rodrigo Agerri currently holds the RYC-2017-23647 fellowship (MCIN/AEI/10.13039/501100011033 and 63.34ESF Investing in your future). We also acknowledge the support of Google's TFRC program.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for basque. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4781–4788.

[16] https://github.com/orai-nlp/low-scaling-laws

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 724–728.

Mikel Artetxe, Itziar Aldabe, Rodrigo Agerri, Olatz Perez-de Viñaspre, and Aitor Soroa. 2022. Does corpus quality really matter for low-resource languages? *arXiv preprint arXiv:2203.08111*.

James Barry, Joachim Wagner, Lauren Cassidy, Alan Cowap, Teresa Lynn, Abigail Walsh, Mícheál J Ó Meachair, and Jennifer Foster. 2021. gabert–an irish language model. *arXiv preprint arXiv:2107.12930*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Casimiro Pio Carrino, Marta R Costa-jussà, and José AR Fonollosa. 2020. Automatic spanish translation of squad dataset for multi-lingual question answering. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5515–5523.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. 2022. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pages 4057–4086. PMLR.

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

Germán Coloma. 2015. Efectos de compensación entre indicadores de la complejidad de los idiomas. Technical report, Serie Documentos de Trabajo.

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.

Miguel Ángel García Cumbreras, Eugenio Martínez Cámara, Julio Villena Román, and Janine García Morera. 2016. Tass 2015–the evolution of the spanish opinion mining systems. *Procesamiento del Lenguaje Natural*, I(56):33–40.

Davis David. 2020. Swahili : News classification dataset. The news version contains both train and test sets.

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.

Asier Gutiérrez Fandiño, Jordi Armengol Estapé, Marc Pàmies, Joan Llop Palao, Joaquin Silveira Ocampo, Casimiro Pio Carrino, Carme Armentano Oller, Carlos Rodriguez Penagos, Aitor Gonzalez Agirre, and Marta Villegas. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104.

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. *arXiv preprint arXiv:2004.09095*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple

subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A Smith. 2021. Probing across time: What does roberta know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842.

Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawendé François D Assise Bissyande, Jacques Klein, Andrey Boytsov, Anne Goujon, and Clément Lefebvre. 2022. Luxembert: Simple and practical data augmentation in language model pretraining for luxembourgish. In *Proceedings of the Language Resources and Evaluation Conference, 2022*, pages 5080–5089.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. 2022a. SwahBERT: Language model of Swahili. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 303–313, Seattle, United States. Association for Computational Linguistics.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jeong Young-Seob. 2022b. Swahbert: Language model of swahili. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 303–313.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte 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.

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.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Ethan Perez and Ian McKenzie. 2022. Inverse scaling prize: Round 1 winners.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26.

Erik F Sang. 2002. Tjong kim (2002)."introduction to the conll-2002 shared task: Language-independent named entity recognition". In *COLING-02: The 6th Conference on Natural Language Learning*.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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 preprint arXiv:1908.08962*.

Gorka Urbizu, Iñaki San Vicente, Xabier Saralegi, Rodrigo Agerri, and Aitor Soroa. 2022. BasqueGLUE: A Natural Language Understanding Benchmark for Basque. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1603–1612, Marseille, France. European Language Resources Association.

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.

Manuel García Vega, Manuel Carlos Díaz-Galiano, Miguel Ángel García Cumbreras, Flor Miriam Plaza del Arco, Arturo Montejo-Ráez, Salud María Jiménez Zafra, Eugenio Martínez Cámara, César Antonio Aguilar, Marco Antonio Sobrevilla Cabezudo, Luis Chiruzzo, et al. 2020. In *IberLEF@ SEPLN*.

David Vilares, Marcos Garcia, and Carlos Gómez-Rodríguez. 2021. Bertinho: Galician bert representations. *arXiv preprint arXiv:2103.13799*.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Yi Tay, and Quoc V Le. 2022b. Inverse scaling can become u-shaped. *arXiv preprint arXiv:2211.02011*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Ves Stoyanov. 2022. Training trajectories of language models across scales.

Rodolfo Zevallos, John Ortega, William Chen, Richard Castro, Nuria Bel, Cesar Toshio, Renzo Venturas, Hilario Aradiel, and Nelsi Melgarejo. 2022. Introducing qubert: A large monolingual corpus and bert model for southern quechua. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 1–13.

Minjia Zhang, Wenhan Wang, Xiaodong Liu, Jianfeng Gao, and Yuxiong He. 2018. Navigating with graph representations for fast and scalable decoding of neural language models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel Bowman. 2021. When do you need billions of words of pretraining data? 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 1112–1125.

A Linguistic Characteristics of Selected Languages.

Table 9 shows the linguistic characteristics of the languages we selected for our experiments, which are Basque (eu), Spanish (es), Swahili (sw) and Finnish (fi). On one hand, we have the language families they belong to, and on the other hand, their complexity in morphology, syntax, verb system and vocabulary according to Coloma (2015).

B Pre-Training Details

We use a cased sub-word vocabulary containing 50K tokens trained with the unigram language model based sub-word segmentation algorithm proposed by Kudo (2018). The vocabularies are learned from each training corpus with a character coverage of 99.95%, to ignore rare characters. Thus, we obtain 3 vocabularies for each language, one for each size of the pre-training corpora (5M, 25M, 125M), which are shared among LMs of different sizes throughout our experiments.

We apply several Maskings to the same sentences, to create different examples from the same text[17], which is a common practice during the preprocessing of the pre-training data. We applied 10 different random maskings to each text and we employed whole-word masking, where whole words are masked instead of the sub-word units. All models were trained on TPUv3-8 machines using the same set of default hyperparameters (Devlin et al., 2019) in all model sizes: a learning rate $1e^{-4}$, $\beta 1$ = 0.9, $\beta 2$ = 0.999, L2 weight decay of 0.01, a learning rate warmup of 10K steps, and training the models for a total of 500K steps with a batch size of 256 and a sequence length of 512. This means that we will be doing many epochs (500/2500/12500K) over the same corpus, a common practice, when there is not an enormous pre-training corpus available, for instance, in the original publication of BERT (Devlin et al., 2019), the model is pre-trained for 40 epochs.

Although the models are trained for the same amount of steps and batch size, the time needed for training each of them is different, with larger models taking more time. We trained all our models using TPUv3-8 machines; in which we trained BERT_{124M} models to 500K steps in 76 hours, BERT_{51M} models in 32 hours and BERT_{16M} models in 10 hours.

[17] I love cats: I love [MASK]; I [MASK] cats; [MASK] love cats

C MLM Evaluation

Table 10 shows the accuracies obtained in the MLM task for each language (Basque, Spanish, Swahili and Finnish), for each model size and corpus size combination. As expected, larger models trained with the biggest corpora yield the best results, and a positive correlation exists between model/corpora size and accuracy in every language we compare. Moreover, results show that in overall, in these low-resource settings, it is preferable to increase the pre-training data over model size. Increasing pre-training data improve results on MLM for all languages, with the exception of the BERT_{16M} model trained with the 125M token dataset. The gain obtained with the smallest BERT_{16M} models as we keep adding training data diminishes, which suggests that performance is reaching a plateau in these models.

On the other hand, increasing the model size only helps once we reach a certain amount of pre-training data. Increasing model size from BERT_{16M} to BERT_{51M} does not improve MLM accuracy for a 5M corpus, suggesting that 3M non-embedding parameters are enough to absorb the knowledge of such a small dataset. However, increasing model size from BERT_{51M} to BERT_{124M} for the same 5M corpus does improve the overall performance for all languages except for Finnish, This might be due to larger language models being more sample-efficient (Kaplan et al., 2020).

Surprisingly, BERT_{51M} outperforms BERT_{124M} consistently across all languages when pre-trained with a 25M corpus; this goes against the intuition of larger models being more sample-efficient. Furthermore, the table shows that a slightly smaller model with more data can outperform a larger model with smaller corpora; every BERT_{51M} model trained with 125M token corpora outperforms BERT_{124M} model trained with 25M tokens.

D Does Overfitting at Pre-training Propagate to Finetuning at Downstream Tasks?

The loss curves in Section 5.1 suggested that some model-dataset size ratios, which have the least data and more model parameters, have been trained for too long, to a degree in which the loss starts to increase significantly.

To analyze if those overfitting issues from when we keep pre-training over and over again on the same training data propagates to the downstream

Language	Language family	Morphology	Syntax	Verb System	Vocabulary
Basque	Language isolate	0.73	0.58	0.77	0.62
Spanish	Romance (Indo-European)	0.64	0.42	0.62	0.69
Swahili	Bantu (Niger-Congo)	0.64	0.42	0.54	0.31
Finnish	Uralic	0.82	0.42	0.46	0.31

Table 9: The four selected languages and their complexity in morphology, syntax, verb system and vocabulary.

MLM_{eu}	5M	25M	125M
$BERT_{16M}$	32.08	38.68	41.56
$BERT_{51M}$	32.42	44.29	50.07
$BERT_{124M}$	34.50	43.46	**53.19**
MLM_{es}	5M	25M	125M
$BERT_{16M}$	39.09	49.06	48.31
$BERT_{51M}$	39.24	53.49	59.04
$BERT_{124M}$	42.45	52.58	**62.00**
MLM_{sw}	5M	25M	125M
$BERT_{16M}$	38.03	45.71	44.98
$BERT_{51M}$	38.08	50.12	55.27
$BERT_{124M}$	40.43	49.06	**58.82**
MLM_{fi}	5M	25M	125M
$BERT_{16M}$	29.43	37.03	37.73
$BERT_{51M}$	28.18	42.07	45.86
$BERT_{124M}$	29.30	41.75	**49.88**

Table 10: Accuracies on MLM for the 4 languages and averages for each model-dataset size. Rows correspond to different Language Model sizes; columns correspond to different corpus sizes employed during pre-training.

tasks once finetuned, here we compare the checkpoints of the models at 100K steps, with the last checkpoint of our models at 500K steps. We did this comparison with 2 models, $BERT_{16}$ and $BERT_{51}$, both of them trained on the smallest corpora (5M words), which are among those with the most pronounced increasing loss curves.

The results for each checkpoint for both models after finetuning on the tasks are shown in Table 12. All in all, the 500K step checkpoints are on a par with the 100K step counterparts, without a clear winner, but definitely equalizing the gap that there is at MLM loss.

Thus, since the decline in loss when kept pretraining does not spread to the downstream tasks, we decide to employ the last checkpoints (500K steps), to evaluate and compare the models at Appendix C and Section 5.2, to avoid adding another variable to the evaluation.

E Optimizing for a Fixed Budget

We have shown that increasing the amount of pretraining data and model size improves their performance. Thus, the conclusion regarding data in low-resourced settings is to use all the data there is available, independently of the model size.

With respect to the model size, however, even if the available corpus size suggests that increasing it improves the performance, there is usually a limited computational budget constraining this. Thus, we need to choose the best model size within our budget. Kaplan et al. (2020) concludes that convergence is inefficient, which means that we obtain optimal performance by training larger models and stopping significantly short of convergence when working with a fixed compute budget.

For this purpose, we compare the $BERT_{51}$ and $BERT_{124}$, pre-trained on a comparable amount of compute. We employed the pre-training corpus of 125M words, and pre-trained $BERT_{51}$ for 500K steps, and $BERT_{124}$ for 200K steps.

The results obtained are shown in Table 11. $BERT_{1}24M$, the model with the most parameters outperforms $BERT_{5}1M$ in most of the tasks: 4/4 for Spanish, 3/4 for Swahili, 2/4 for Basque and 2/4 for Finnish. These results agree with the claim of Kaplan et al. (2020) that *convergence being inefficient*. However, since there is not a big gap in the results, other factors might be also considered. For example, an undertrained $BERT_{124M}$ model has more room for improvement with further pretraining, while $BERT_{51M}$ is cheaper and faster to finetune and deploy.

F Comparison with SotA on Downstream Tasks

In Table 13 we compare the results of our $BERT_{124M}$ trained over the corpora of 125M words, the baselines of Flair, and mBERT with the current state-of-the-art results on each language and task. We improve SotA results for NERC, topic

		BERT_{51M}	BERT_{124M}
	Steps	500K	200K
	FLOPs	1.002E+19	1.347E+19
eu	NERC	82.98±0.1	**83.67**±0.3
	Topic	74.87±0.4	**75.49**±0.3
	SA	**69.40**±0.9	68.43±0.8
	QNLI	**74.18**±1.6	72.49±2.9
es	NERC	86.34±0.7	**86.47**±0.7
	Topic	95.91±0.4	**95.95**±0.1
	SA	39.89±0.0	**42.79**±1.1
	QNLI	78.00±0.6	**79.65**±0.4
sw	NERC	91.63±0.1	**91.66**±0.2
	Topic	92.88±0.2	**93.07**±0.1
	SA	77.07±0.0	**77.60**±1.1
	QNLI	**63.87**±1.5	63.63±0.9
fi	NERC	**87.02**±0.2	86.86±0.7
	Topic	89.61±0.3	**89.63**±0.2
	SA	92.58±0.0	**92.63**±0.5
	QNLI	**57.30**±4.3	56.35±7.6

Table 11: Results of BERT_{51M} and BERT_{124M} models trained with comparable computational budget (1E+19 FLOPs), for 500K and 200K steps respectively.

classification[18] and sentiment analysis for Swahili, and obtain similar results for topic classification for Finnish.

G MLM Loss Plots Zoomed in for BERT_{16M}

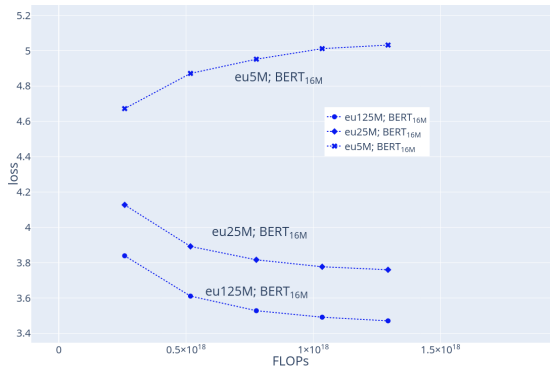

Figure 5: MLM loss and FLOPs for BERT_{16M} models for Basque, zoomed in from Figure 1.

Since the loss curve lines for BERT_{16M} for the corpora of 25M and 125M tokens are hard to see in Figures 1, 2, 3 and 4, we zoomed in on them in the figures 5, 6, 7 and 8 respectively.

[18]Our model is finetuned with a subset of the dataset

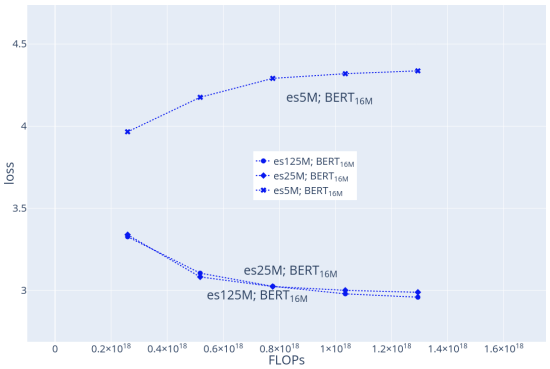

Figure 6: MLM loss and FLOPs for BERT_{16M} models for Spanish, zoomed in from Figure 2.

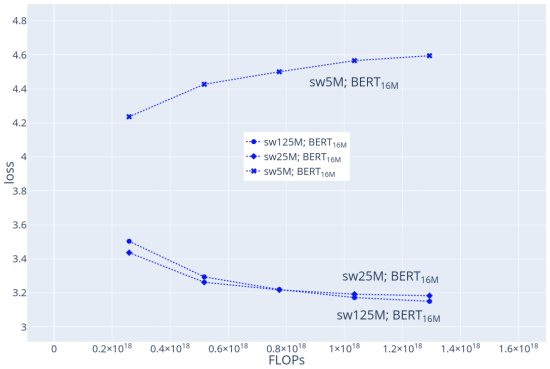

Figure 7: MLM loss and FLOPs for BERT_{16M} models for Swahili, zoomed in from Figure 3.

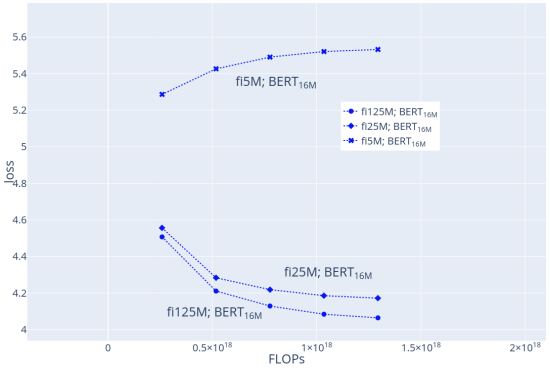

Figure 8: MLM loss and FLOPs for BERT_{16M} models for Finnish, zoomed in from Figure 4.

H Takeaways from Scaling Laws for Low-Resource Settings

In Tables 14-17 we compare our results to the predictions of previous scaling laws from Kaplan et al. (2020)[19] and Hoffmann et al. (2022)[20].

Tables 14-17 show the estimates of Kaplan et al. (2020) do not hold in this low-resource setting, by several magnitudes of order. Table 15 shows how

[19]$a = 0.73$ and $b = 0.27$

[20]$a = 0.5$ and $b = 0.5$

		BERT$_{16M}$		BERT$_{51M}$	
Lang	Task	100K step	500K step	100K step	500K step
eu	MLM loss	**4.6724**	5.0323	**5.9511**	8.4830
	MLM acc	31.56	**32.08**	**33.60**	32.42
	NERC	**64.75**±0.6	63.90±0.5	70.13±0.4	**70.14**±0.4
	Topic	**68.56**±0.4	68.00±0.6	**70.18**±0.6	69.98±0.6
	SA	67.16±0.5	**67.80**±0.5	**67.32**±0.4	67.00±1.0
	QNLI	**68.20**±1.2	68.10±2.3	64.64±1.6	**65.06**±0.8
es	MLM loss	**3.9658**	4.3367	**5.8144**	8.2425
	MLM acc	**40.15**	39.91	**39.27**	38.93
	NERC	75.84±0.4	**76.57**±0.3	**81.11**±0.3	80.43±0.4
	Topic	**94.69**±0.4	94.54±0.3	**94.89**±0.3	**94.89**±0.3
	SA	37.26±0.8	**37.67**±1.5	35.68±3.9	**36.05**±2.1
	QNLI	**65.38**±1.0	65.01±0.6	66.34±1.6	**67.00**±1.8
sw	MLM loss	**4.2363**	4.5952	**5.9836**	8.4719
	MLM acc	37.64	**38.03**	**38.70**	38.08
	NERC	85.92±0.5	**86.36**±0.2	**89.05**±0.2	88.74±0.2
	Topic	91.28±0.3	**91.64**±0.3	91.85±0.4	**92.12**±0.2
	SA	71.31±0.5	**71.52**±0.6	**71.01**±0.6	70.49±0.7
	QNLI	60.29±0.9	**62.80**±1.1	**62.88**±0.9	62.27±1.7
fi	MLM loss	**5.2866**	5.5322	**6.5559**	8.9016
	MLM acc	28.76	**29.43**	**29.52**	28.18
	NERC	76.47±0.3	**76.82**±0.3	**80.19**±0.2	79.73±0.2
	Topic	**88.53**±0.1	88.15±0.1	88.45±0.1	**88.53**±0.3
	SA	**89.95**±0.1	89.69±0.2	**90.50**±0.2	89.61±0.6
	QNLI	**51.51**±1.1	51.49±0.9	52.37±0.7	**54.28**±2.5

Table 12: Comparison on downstream tasks of different checkpoints (100-500K step) of the models that showed over-fitting issues during pre-training due to over-parametrizing.

		BERT$_{124M}$	Flair	mBERT	SotA	
eu	NERC	84.58±0.2	82.13±0.4	79.39±1.0	**86.98**±0.4	roberta-euscrawl-l(Artetxe et al., 2022)
	Topic	76.06±0.4	67.89±0.3	70.57±0.5	**86.51**±0.4	ElhBERTeu (Urbizu et al., 2022)
	SA	68.91±0.5	68.17±0.3	67.34±0.7	**70.87**±0.5	Berteus (Agerri et al., 2020)
	QNLI	74.09±1.7	48.66±5.2	**78.48**±1.9	76.04±1.5	ElhBERTeu (Urbizu et al., 2022)
es	NERC	87.28±0.3	87.09±0.3	87.21±0.4	**88.51**	roBerta-b(Gutiérrez Fandiño et al., 2022)
	Topic	96.27±0.3	94.08±0.4	95.92±0.6	**97.14**	BETO(Gutiérrez Fandiño et al., 2022)
	SA	43.27±1.1	34.73±3.0	39.21±1.8	**49.80**	Vega et al. (2020)
	QNLI	81.10±0.7	56.42±0.6	**83.92**±0.2	82.02	roberta-l(Gutiérrez Fandiño et al., 2022)
sw	NERC	**92.09**±0.2	92.04±0.1	91.17±0.1	88.60	swahBERT (Martin et al., 2022b)
	Topic	**93.07**±0.2	91.83±0.2	91.52±0.2	90.90	swahBERT (Martin et al., 2022b)
	SA	**79.04**±0.7	73.60±0.5	69.17±1.2	71.12	swahBERT (Martin et al., 2022b)
	QNLI	63.34±1.4	52.82±2.1	63.48±1.1	**64.72**±0.4	swahBERT(our evaluation)
fi	NERC	88.99±0.2	84.76±0.4	88.87±0.4	**92.40**±0.1	finBERT(Virtanen et al., 2019)
	Topic	90.14±0.1	86.58±0.7	88.16±0.3	**90.57**±0.2	finBERT(Virtanen et al., 2019)
	SA	94.38±0.3	89.74±0.5	88.05±0.7	**95.61**±0.3	finBERT(our evaluation)
	QNLI	**58.56**±1.1	51.54±1.2	52.79±3.0	57.18±2.6	finBERT(our evaluation)

Table 13: Results of BERT$_{124}$ model pre-trained with the biggest dataset (125M words), the Flair baseline pre-trained with the same corpora, mBERT and current SOTA.

	Optimal FLOPs		
Dataset	Kaplan	Hoffman	Ours
125M	9.74E+29	1.56E+16	3.37E+19
25M	2.51E+27	6.25E+14	5.39E+18
5M	6.47E+24	2.50E+13	1.35E+18

Table 14: Optimal FLOPs for each dataset size.

	Optimal model size		
Datasets	Kaplan	Hoffman	Ours
125M	7.79E+21	1.25E+08	>8.56E+07
25M	1.00E+20	2.50E+07	>8.56E+07
5M	1.29E+18	5.00E+06	8.56E+07

Table 15: Optimal model size in parameters for each dataset.

	Optimal FLOPs		
Model	Kaplan	Hoffman	Ours
BERT_{124M}	7.36E+10	7.33E+15	3.37E+19
BERT_{51M}	1.40E+10	6.49E+14	1.00E+19
BERT_{16M}	8.47E+08	1.08E+13	1.29E+18

Table 16: Optimal FLOPs for each model size.

	Optimal dataset size		
Model	Kaplan	Hoffman	Ours
BERT_{124M}	8.59E+02	8.56E+07	>1.25E+08
BERT_{51M}	5.48E+02	2.55E+07	>1.25E+08
BERT_{16M}	2.57E+02	3.29E+06	1.25E+08

Table 17: Optimal dataset size in tokens for each model size.

Hoffmann estimations for optimal model size are not that far from our results, but Table 17 suggests that the optimal dataset size required to train small language models is around an order of magnitude higger than what the scaling laws of Hoffmann et al. predict. Furthermore, Tables 14 and 16 show that the optimal FLOPs needed for those models are a few orders of magnitude higger than predicted by Hoffmann et al., where models are trained for a single epoch, which is clearly not optimal in low-resource settings.

All in all, we can underline the following take-aways for NLP practitioners working on LMs in low-resource settings:

- Use as much text as available.
- Pretraining for several (100s) epochs is clearly beneficial.
- Given a fixed computational budget, it is better to train big models instead of using the computational power to compute more model updates in smaller models.
- For a dataset of 125M words: BERT_{124M}, trained for at least 3.37E+19 FLOPs[21] is recommended.
- For a 25M dataset: BERT_{124M}, trained for 5.39E+18 FLOPs[22] is recommended, but a BERT_{51M} model trained for 3.01E+18 FLOPs[23], obtains similar results, and it is lighter for finetuning and inference.
- For a 5M dataset: BERT_{124M}, trained for 1.35E+18 FLOPs[24] or less is recommended, but a BERT_{51M} model trained for 7.01E+17 FLOPs[25], obtains similar results, which is lighter for finetuning and inference.

[21] 500K steps with batch=256 and sequence-length=512
[22] 80K steps with batch=256 and sequence-length=512
[23] 150K steps with batch=256 and sequence-length=512
[24] 20K steps with batch=256 and sequence-length=512
[25] 35K steps with batch=256 and sequence-length=512

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations (after conclusions)

☒ A2. Did you discuss any potential risks of your work?
We are not aware of any potential risks of our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and introduction (1).

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3 Experimental Setup

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3 Experimental Setup and 4 Evaluation Settings

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

5 Results and Appendix

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3 Experimental Setup, 5 Results and Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3 Experimental Setup and Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3 Experimental Setup and 5 Results

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3 Experimental Setup, 4.6 Systems and Baselines and 5 Results

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.