Scaling behavior of Encoder Language Models in Low-Resource Settings

Ruan Visser[1]*, Trienko Grobler[1], and Marcel Dunaiski[1]

Department of Computer Science, Stellenbosch University, Stellenbosch, South Africa
ruanvisser101@gmail.com, tlgrobler@sun.ac.za, marceldunaiski@sun.ac.za

Abstract. Pretraining language models for low-resource languages poses significant challenges due to scarce and poor-quality data, a lack of comprehensive evaluation benchmarks, and often limited computational resources. Research on compute-optimal language modeling typically focuses on scaling up decoder language models efficiently for high-resource languages. While some studies have investigated the down-scaling of encoder language models for low-resource languages, they often prioritize optimizing for computational constraints rather than pretraining text volume constraints. We address this research gap by analyzing the scaling behaviors of encoder language models which use the Replace Token Detection (RTD) and Masked Language Modeling (MLM) objectives under limited pretraining text volumes.
By downsampling three different high-resource languages (English, French, Korean) and two low-resource languages (Xhosa and Swahili), we simulate varying degrees of data scarcity and evaluate downstream performance using established benchmarks such as the GLUE benchmark for English, FLUE for French, KLUE for Korean, and MasakhaNEWS for Xhosa and Swahili. Our findings demonstrate that optimal MLM accuracy scales logarithmically with increasing pretraining text volume across these diverse languages. Additionally, our results show that RTD models consistently outperform MLM models in low-resource scenarios, achieving superior downstream performance with pretraining text volumes smaller than 1000MB for downsampled high-resource languages. However, we find that RTD performs worse than MLM for Xhosa and Swahili. We also find that dynamic masking significantly improves MLM accuracy in these settings. Furthermore, our results show that smaller models are more effective for smaller pretraining text volumes, highlighting the importance of adjusting model size according to data availability in order to maximize performance and efficiency.

1 Introduction

Recent work on compute optimal language modeling has enabled researchers to train better performing large language models (LLMs) while minimizing computational requirements. Notable studies such as Hoffmann et al. [11] and Kaplan

* Corresponding author

et al. [13] have established guidelines for optimizing resource allocation based on data and computational constraints. While research on optimizing large-scale language model training often produces relevant findings that are applicable to low-resource settings, these studies are primarily concerned with efficiently scaling up models [13,11,18].

Researchers that focus on low-resource languages typically face more severe resource constraints, such as restricted computational resources and a lack of good-quality textual data, which led to an increased focus on more efficient models or methods. Various approaches have been explored to improve the pretraining efficiency of language models, including language specific model or data augmentation [19,22,17], improvements to the attention mechanisms [12,9], and the use of more efficient pretraining objectives. Among these, the choice of pretraining objective is particularly significant. For example, BERT-base [6] which is trained using the Masked Language Modeling (MLM) task was able to significantly outperform GPT [21], which was trained using Causal Language Modeling (CLM), on the General Language Understanding Evaluation (GLUE) benchmark [26]. Furthermore, the Replace Token Detection (RTD) objective proposed by Clark et al. [3] specifically emphasizes pretraining efficiency, with a RTD model obtaining results comparable to a MLM model with only 1/4 of the compute resources.

More efficient training objectives may lead to more advantageous scaling behaviors in lower-resource settings when both compute and data volumes are limited. However, research on compute optimal pretraining often focuses solely on CLM models. A handful of studies have, however, explored the scaling behavior of MLM models in low-resource settings. Notably, Urbizu et al. [24] and Deshpande et al. [5] identified a significant divergence from the original scaling laws proposed by Hoffmann et al. [11] and Kaplan et al. [13] when applied to encoder models in low-resource settings. These findings suggest that the established scaling laws may not be directly applicable to encoder models trained using limited data volumes.

With this paper, we provide insights and information useful for making informed model decisions during the development of efficient low-resource language models, emphasizing strategies to overcome computational barriers. To this end, we analyze the scaling behaviors of encoder language models in low-resource settings. However, pretraining low-resource language models presents several challenges, including the large computational requirements, limited pretraining data, data quality issues, and limited evaluation datasets. Similar to prior work [28,5,16,24], we circumvent the lack of quality pretraining data and comprehensive downstream benchmarks for low-resource languages by downsampling high-resource languages. More specifically, we downsample three languages individually, namely English, French, and Korean, since they have distinct linguistic characteristics and comparable downstream benchmarks. This methodology allows us to simulate the data scarcity of low-resource languages while allowing us to evaluate downstream performance using established benchmark datasets such as the General Language Understanding Evaluation (GLUE). Furthermore,

we evaluate the scaling behaviors of two pretraining objectives, namely, Replace Token Detection (RTD) and Masked Language Modeling (MLM). To further validate our findings, we also downsample actual low-resource languages, Swahili and Xhosa, and evaluate the resulting models on the MasakhaNEWS [1] downstream dataset.

Similar to previous work that shows a smooth power law relationship between optimal model performance and CLM validation loss [13,11,18], we find that MLM accuracy scales logarithmically as pretraining text volume is increased for optimally trained BERT models. We define "optimal models" as the model which achieves the best performance for a given pretraining text volume, considering both the model size and the number of pretraining steps. Our analyses indicate that this logarithmic scaling property generalizes across five diverse languages as text volumes increase. Consequently, the optimal models identified in this paper can be directly used by language modeling practitioners when pretraining future low-resource language models without the need to execute costly experiments to find the optimal model parameters for their use case.

Additionally, our results highlight the importance of selecting the appropriate pretraining objective when training in low-resource settings. While the more efficient RTD pretraining objective generally leads to improved downstream performance for downsampled high-resource languages, the more commonly used MLM objective performs best for downsampled low-resource languages. Specifically, for models trained with less than 1000MB of English, French, or Korean pretraining data, the more compute efficient RTD objective outperforms equivalent MLM models in most downstream tasks. However, for low-resource languages, such as Xhosa and Swahili, we observe the opposite trend with MLM models consistently outperforming RTD models on the MasakhaNEWS dataset. Furthermore, we find that MLM improvements such as dynamic masking are effective in lower-resource settings. For example, we find that a BERT model which uses dynamic masking is able to achieve better results for some downstream tasks compared to the original BERT while using almost two orders of magnitude less input data.

2 Related Work

2.1 Scaling Language Models

Language models have improved rapidly in recent years, driven primarily by increases in model sizes, the amount of training data used, and the available compute. As a result, to train competitive models nowadays requires significant computational resources which makes it increasingly difficult for researchers to study and small organizations to experiment with LLMs. Many approaches exist that attempt to identify optimal scaling strategies to train larger and better performing language models under constrained resources scenarios [13,11,18], which may guide researchers to more efficiently pretrain future models.

Kaplan et al. [13] conducted the first study on compute optimal LLMs and found a power law relationship between the number of parameters within a

language model and its test loss. They found that for every 10 fold increase in compute, the number of training tokens should be increased by a factor of 1.8 and model size should be increased by a factor of 5.5. Additionally, Kaplan et al. [13] found that models trained until convergence are not compute optimal. Alternatively, Hoffmann et al. [11] argue that the recommendations created by Kaplan et al. [13] result in significantly undertrained models and suggest that model size and number of training tokens should be scaled equally. They show the efficacy of this scaling behavior prediction by training Chinchilla, a model which is able to significantly outperform Gopher, a 280 billion parameters LLM, by training on 4 times more data while using a similar amount of compute and 1/4 the number of parameters. More recently Muennighoff et al. [18] noted that, according to the scaling law proposed by Hoffmann et al. [11], the current trend of rapidly increasing sizes of LLMs will soon lead to a demand for training data volumes exceeding the total amount of publicly available text. In light of this issue, they analyzed the effect of training language models for multiple epochs, rather than the single epoch which has become standard practice for LLMs. They found that the loss of a model trained once on a full training dataset is similar to that of a model trained for 4 epochs on 1/4 of the training dataset. Consequently, they recommend training for multiple epochs to extract a higher quality signal from the data. However, they also observe that the improvements obtained by training for additional epochs eventually reduces to zero.

Instead of following the prevalent trend of expanding decoder models, Deshpande et al. [5] assessed the efficacy of pretraining encoder models using the MLM training objective at a smaller scale. To better analyze the impact of pretraining smaller models, they reduced the languages' complexity by filtering out pretraining data that contains words outside of a simplified child-directed speech corpus. They find that models with as few as 1.25M parameters benefit from pretraining. Additionally, they observed a discontinuity in the FLOPs-perplexity relationship, with a sudden shift in the exponent value in the low-compute region, diverging from previous scaling law observations.

Urbizu et al. [24] also study the scaling behaviors of MLM models for low-resource scenarios. Instead of training on simplified English, they trained multiple monolingual models across four languages, namely Basque, Spanish, Swahili, and Finnish. Despite employing the more efficient MLM objective and training for multiple training epochs, compared to a single-epoch and the CLM objective approach used by Hoffmann et al. [11], they conclude that compute optimal models require larger data volumes than what is suggested by the scaling laws created by Hoffmann et al. [11]. The work by Urbizu et al. [24] and Deshpande et al. [5] reveals that the conventional scaling laws, while instrumental in guiding the development of large decoder models, may not be fully applicable for optimizing encoder models in low-resource settings.

2.2 Pretraining objective

CLM is an unidirectional pretraining objective which learns from text data by incrementally predicting next tokens based on previous tokens within a context

[21]. Although this objective is effective for training models that generate text, models trained using CLM can only learn using prior tokens, effectively ignoring signals from later tokens. This objective is used to train decoder models such as GPT [21], which was the state-of-the-art model at the time. CLM-based models remain the most often used LLMs for evaluating the scaling behaviors [13,11,18].

Devlin et al. [6] introduced Masked Language Modelling (MLM), a deeply bidirectional objective that enables every token to attend to all other tokens in a sequence. This objective was used to pretrain BERT, an encoder model that outperforms GPT on various language understanding tasks in the GLUE benchmark. Devlin et al. [6] believe that the deeply bidirectional nature of the MLM objective is the reason for the improved performance on language understanding tasks.

Clark et al. [3] introduced the more efficient Replace Token Detection (RTD) objective with the ELECTRA model. In the RTD objective, randomly selected tokens are substituted with closely matching generated alternatives. ELECTRA is then tasked with determining whether each token was generated or was part of the original input sequence. Unlike BERT, which predicts the identity of 15% of tokens, ELECTRA has to determine the correctness of every token. Clark et al. [3] studied the impact of the RTD compared to MLM pretraining objective using the same model size, data volume, and compute budget and found that ELECTRA substantially outperforms BERT on GLUE. They argue that the enhanced performance is primarily due to the increased number of decisions made per sequence in the RTD compared to the MLM task.

2.3 Multilingual models

Multilingual models such as mBERT [6] and XLM-R [4] have emerged as a viable options for improving performance for low-resource languages by leveraging the shared linguistic features of multiple languages. These models are pretrained on diverse sets of languages, enabling them to transfer knowledge from high to low-resource languages, thus improving performance in multilingual contexts.

Language-specific performance of multilingual models can be further improved with continual pretraining, by further training a pretrained multilingual model on a specific language or set of languages to adapt it more closely to the target languages. For instance, AfroXLM-R [2] was continually pretrained using an XLM-R model on 17 African languages. This approach allows the model to retain general language knowledge while also refining its performance on specific languages.

3 Methodology

3.1 Language sampling

In contrast to most studies that analyze scaling behaviors of language models, we pretrained models on multiple diverse languages individually, namely English,

French, and Korean. English and French have similar scripts and a significant overlap in vocabulary [8], while Korean differs significantly in both vocabulary and language symbols. This diversity allows us to evaluate whether language characteristics affect the scaling behavior of language models or whether language independent patterns may be found. Additionally, we extend our analysis to include two low-resource languages, namely, Xhosa and Swahili.

We sampled seven data volumes ranging from 2M to 200M tokens from the mC4 [27] and Oscar [20] text corpora. The specific data volumes are 12MB, 25MB, 50MB, 100MB, 250MB, 500MB, and 1GB, which results in 7 distinct data volumes for both mC4 and Oscar.[1] Our sampling strategy involved splitting the entire common crawl datasets into 1MB chunks and randomly selecting chucks until the desired data volume was reached. The 100MB-1GB volumes represent the amount of data commonly found for low-resource languages in the common crawl dataset. For instance, Samoan has 245MB of textual data and Lao has 641MB. Furthermore, these volumes align with the 10 million to 100 million words which Zhang et al. [28] suggest could enable language models to reliably acquire syntax comprehension capabilities. The smaller 12MB-50MB volumes represent the volumes that can be obtained when scraping several websites. While Zhang et al. [28] argue that these volumes might not be adequate for RoBERTa-base [15] models to learn syntax, we examine if smaller models or the RTD objective can better enable syntactical learning from these volumes.

3.2 Model Scaling

We assess the scaling behavior of two distinct model objectives: RTD and MLM. For each objective, we trained two dedicated model sizes. In addition, we also included the MLM generators from each RTD model which results in a total of four MLM models: BERT-xsmall (3M), BERT-small (12M), BERT-medium (40M), BERT-base (110M) and two RTD models: ELECTRA-small (12M) and ELECTRA-base (110M). To assess the benefit of additional pretraining compute, we evaluated both the pretraining and fine-tuning performance of model checkpoints at specific training steps of 20 000, 100 000, 300 000, as well as the final fully-trained model.

3.3 Pretraining

We pretrained all models using the standard base and small pretraining hyperparameters, while for the generators we used the hyperparameters outlined by Clark et al. [3]. However, to reduce computational requirements, we decreased sequence lengths from 512 to 128 which reflects the computational limitations faced by many practitioners who train on low-resource languages.

To evaluate the pretraining performance of our model checkpoints we held out 5MB of pretraining text as a validation set for each language. We use MLM

[1] Only mC4 contains sufficient Swahili and Xhosa text. For Xhosa, we only sampled up to 100MB as the total available text volume is less than 250MB.

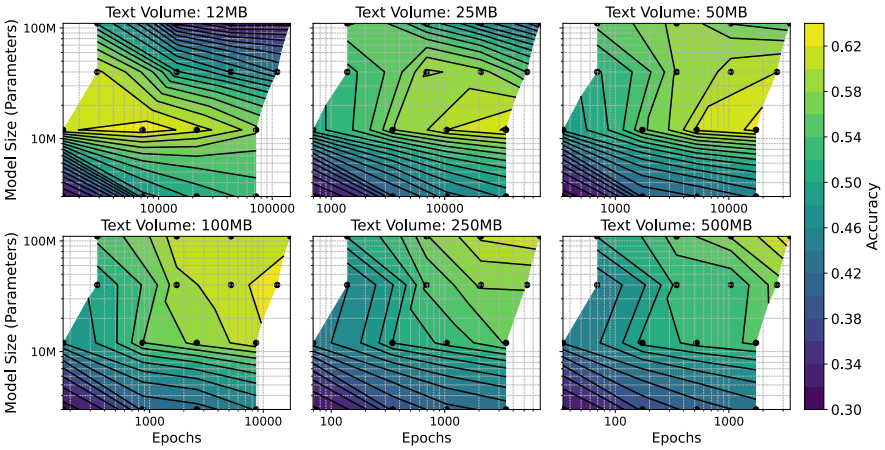

Fig. 1. Contour plots of the MLM accuracy of models using varying amounts of epochs and parameters. Each subplot shows the performance of 16 model checkpoints trained with specific data volumes of English Oscar text.

accuracy when evaluating BERT models and RTD accuracy for ELECTRA models. MLM accuracy corresponds with the number of times the model correctly predicts masked words in a text, while RTD accuracy corresponds with the number of times the model correctly identifies replaced tokens. We determine the optimal model for each pretraining text volume by selecting the model which has the best pretraining performance across all models sizes and checkpoints.

3.4 Fine-tuning

We evaluated the language understanding performance of each model by fine-tuning them on specific language understanding tasks. For English, we used the Generalized Language Understanding Evaluation (GLUE) benchmark which includes a wide variety of classification tasks including linguistic acceptability (CoLA), sentiment classification (SST-2), sentence similarity (STS-B), paraphrase identification (MRPC), and textual entailment (RTE).

We also used tasks from the Korean and French GLUE alternatives, named KLUE and FLUE respectively. Specifically, we evaluated our French and Korean models using the paraphrase identification (PAWS-X) and textual entailment (NLI) tasks, respectively. For Xhosa and Swahili we evaluated the downstream performance using the MasakhaNEWS news topic classification datasets.

When fine-tuning both RTD and MLM models, we applied the model size-dependent hyperparameters outlined by Clark et al. [3]. Tables 1 and 3 in the Appendix give all relevant hyperparameters for pretraining and fine-tuning, respectively. Additionally, Section 5.1 details our hyperparameter tuning process.

4 Results

4.1 Pretraining Results

In our analysis "optimal model" refers to the model that achieves the best performance for a given pretraining text volume, taking into account both the model size and the pretraining step count. Figure 8 depicts the increase in the number of times input data is reused as training duration increases. From this figure, we can see that the optimal BERT-base models reuse the input data between 1 000 and 10 000 times, independent of the input text volume available. Furthermore, Figure 1 shows the MLM validation performance of our English models when varying both the size of models and number of training epochs. From these plots, we observe that the optimal model MLM performances are generally obtained after 5 000 epochs but only when the chosen model size is not too large for the pretraining text volume. This contrasts with recent research findings on scaling decoder language models which suggest that training for more than several epochs may be computationally wasteful [18]. Furthermore, these results also do not agree with other work that recommends against the reuse of data when pretraining models [10]. Although many well-known encoder models are trained for hundreds of epochs [6,3,15], it is typically not recommended to repeat data thousands of times.

Figure 2 gives the optimal model MLM validation accuracy achieved over increasing pretraining text volumes for the three different languages as well as the different common crawl datasets sampled for the pretraining text. When comparing models that are trained using mC4 and Oscar, we do not observe any significant performance differences. Furthermore, we observe similar data scaling improvements across all three languages, with the optimal model performance of each language improving by approximately 25 percentage points between our smallest and largest text volumes.

Regarding optimal model sizes, the results show that models with 12M parameters perform the best for text volumes smaller than 50MB. As the pretraining text volume increases, larger models with greater capacity tend to perform better. Accordingly, for text volumes between 50MB and 100MB, 40M models are preferred, while base-sized models (110M) achieve the highest accuracy for text volumes larger than 100MB. These optimal model sizes seem to generalize well across the different languages and pretraining datasets.

To determine the relationship between pretraining text volumes and optimal model performance, we adopted a similar approach to Deshpande et al. [5]. Specifically, we fit the data to a logarithmic function of the form $y = a\log(x) + b$ and found that for each pretraining dataset, the optimal model MLM performance scales logarithmically with R-squared values of over 0.96. See Figure 6 and Table 4 in the Appendix for the fitted logarithmic curves and the fitted parameters.

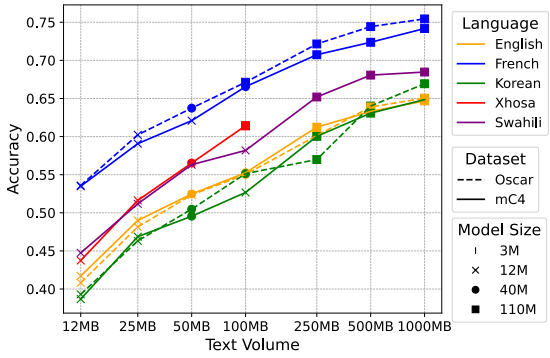

Fig. 2. The optimal model MLM accuracy for increasing pretraining text volumes. The line color and style respectively indicate the target language and common crawl dataset sampled for the pretraining text. The marker type shows the optimal model size for the corresponding pretraining data volumes.

4.2 Fine-tuning results

Figure 3 shows the downstream paraphrase identification (MRPC) performance of the same English models considered in Figure 1. We observe that the exact model checkpoints which obtain the best pretraining performance did not necessarily perform the best when fine-tuned. However, similar performance trends as in Figure 1 can be seen when the model size, compute, and pretraining text volumes are increased.

A mismatch between pretraining and fine-tuning is not uncommon. Tay et al. [23] also observed that pretraining performance can be a misleading indicator of downstream performance. Conversely, Deshpande et al. [5] found that there is a strong correlation between pretraining and fine-tuning performance when training BERT models in low-resource settings. Alternatively, the inherent variability of fine-tuning performance can distort results [7].

Clark et al. [3] find that ELECTRA models are significantly more compute-efficient than BERT models. In Figure 4, we compare the sample efficiency of differently-sized ELECTRA and BERT models by indicating the downstream MRPC performance of several pretraining checkpoints (20 000, 100 000, 300 000, and fully trained) and the corresponding FLOPs used. ELECTRA and BERT models' performances are shown in blue and red respectively. Although ELECTRA uses more FLOPs per pretraining step due to the addition of the models' generator component, it consistently outperforms BERT across almost all data volumes for both small and base-sized models under similar compute constraints. We also observe that small models generally achieve better performance per FLOP for both ELECTRA and BERT. However, this could be due to the characteristics of the MRPC dataset and might not be generally true. Notably, the fully-trained ELECTRA-small checkpoints exhibit the best performance across most data volumes. ELECTRA-small trained on only 100MB of pretraining data

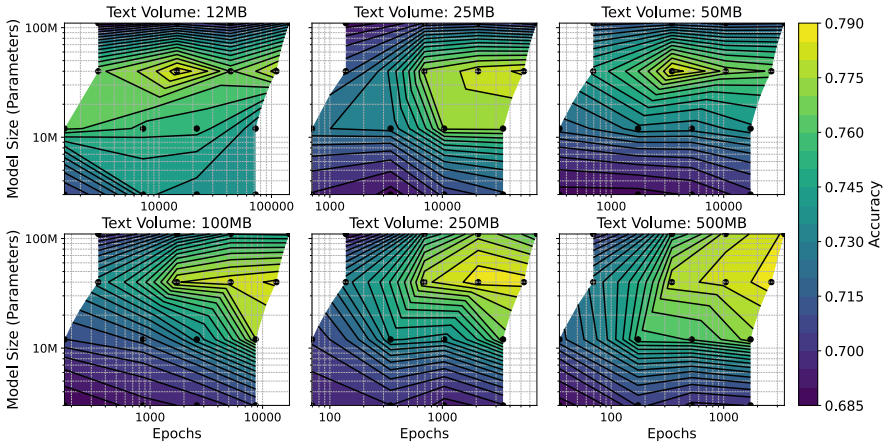

Fig. 3. Contour plots of the downstream MRPC accuracy of models using varying amounts of epochs and parameters. Each subplot indicates the performance of 16 model checkpoints trained using different pretraining text volumes of English Oscar text.

manages to outperform BERT-base by Devlin et al. [6], which was trained on 16GB of text.

Figure 5 shows the downstream performance obtained by the English ELECTRA and BERT models across 5 different English downstream tasks. We observe that ELECTRA consistently and significantly outperforms BERT across STS-B (sentence similarity), RTE (textual entailment), CoLA (grammatical acceptability), and MRPC (paraphrase detection). Moreover, the optimal ELECTRA model trained on 100MB of pretraining data outperforms the optimal BERT model trained on 1000MB of text for three tasks (STS-B, RTE, MRPC). However, for sentiment classification (SST-2) BERT marginally outperforms ELECTRA.

We also find that our models outperform the original BERT-base [6] on both SST-2 and MRPC, while for STS-B the ELECTRA models obtain performances similar to BERT-base. These results indicate both the range of expected performance that can be achieved with limited text volumes as well as the improvement gains that can be obtained with more efficient model architectures such as RTD and dynamic masking.

In terms of model size for BERT models we find similar tends as in Figure 2 across most tasks, with small models being generally preferred for pretraining text volumes up to 50MB, 40M parameter models for text volumes between 50-100MB, and base models for text volumes greater than 250MB. However, smaller BERT models are generally preferred for the RTE downstream task. These results contradict the findings of Urbizu et al. [24] who observed that their largest model, a base-sized model, tends to obtain the best downstream performance even when pretrained using relatively small text volumes of 5M and 25M tokens. However, they use the same hyperparameters typically used to

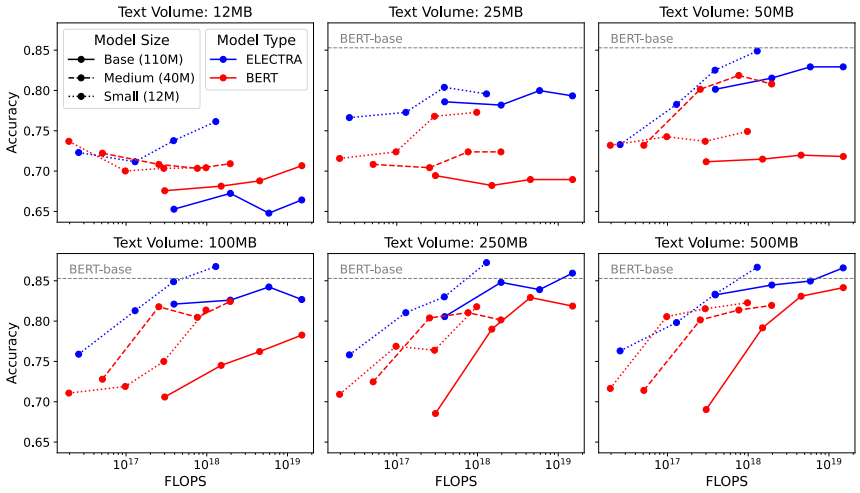

Fig. 4. Downstream MRPC accuracies plotted against the amount of FLOPs used across different model types (line colour), model sizes (line style), and pretraining step counts. Each subplot shows the achieved model accuracies when pretrained on different pretraining text volumes of English Oscar text.

train BERT-base when both pretraining and fine-tuning all their models, which likely negatively affects the performances of smaller models.

Figure 7 depicts the downstream performance of the low-resource languages, Xhosa and Swahili. We observe that models pretrained with only 12MB of Xhosa text data significantly outperform XLM-R-base. Moreover, BERT models trained on 25MB or more of pretraining text can outperform AfroXLM-R-base. For Swahili, while BERT models pretrained with a relatively small amount of text data can surpass XLM-R-base, its performance still falls short of AfroXLM-R-base.

In contrast to most higher-resource language downstream results, we find that BERT tends to outperform ELECTRA across all text volumes for Xhosa and Swahili news classification. This finding aligns with Visser et al. [25], who also report that BERT outperforms ELECTRA across multiple African languages for three different downstream tasks, including NER, POS, and news classification. This discrepancy in performance may be attributed to the poor data quality of textual data available in common crawl datasets for these languages. Both Kreutzer et al. [14] and Visser et al. [25] note the low-quality of low-resource common crawl text, with Visser et al. [25] stating that the majority of sentences in the Zulu mC4 dataset do not contain common Zulu words.

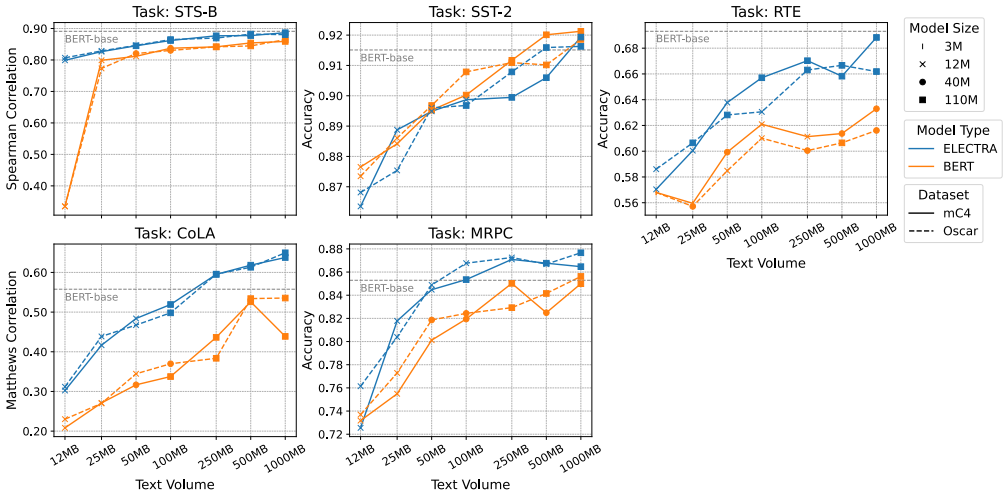

Fig. 5. Downstream performance of optimal BERT and ELECTRA models trained using varying pretraining text volumes. The line colors indicate the used pretraining objective, while the line style corresponds with the dataset sampled for pretraining. Each subplot shows the optimal model performances for a specific downstream task.

5 Conclusion

We analyzed the scaling behaviors of BERT and ELECTRA models in low-resource settings to create practical guidelines for language modeling practitioners working with low-resource languages. While Urbizu et al. [24] also evaluate the scaling laws in low-resource settings, they suggest training with larger text volumes and larger models, both of which are typically unattainable for low-resource practitioners. In contrast, we focused on evaluating the optimal model size and compute requirements for fixed pretraining text volumes.

We found that optimal MLM accuracy scales logarithmically as pretraining text volume increases. Specifically, BERT models with 12M parameters are optimal for text volumes smaller than 50MB, 40M parameter models are preferred for text volumes between 50MB and 100MB, and 110M parameter models perform best for text volumes greater than 100MB. These trends are consistent across English, French, and Korean, as well as two different pretraining datasets for each language, suggesting broad applicability of our observations.

We also found that the more efficient RTD objective leads to significantly improved downstream performances across multiple languages and different downstream tasks. However, our downstream results for Xhosa and Swahili indicate that BERT consistently outperforms ELECTRA across all text volumes. These results emphasise the importance of selecting the appropriate pretraining objective in low-resource settings. Furthermore, the dynamic masking technique introduced by Liu et al. [15] further enhances BERT models, allowing it to out-

perform the original BERT-base [6] in certain downstream tasks while using only a small fraction of the pretraining text.

Limitations

Due to computational constraints, we only tuned the learning rate using a basic grid search. More fine-grained and task-specific parameter tuning could result in clearer scaling behaviors. Future research should validate the generalizability of our results by evaluating models trained on additional actual low-resource languages for which appropriate and comparable downstream test datasets exist.

Acknowledgements

This work was supported by Google's TPU Research Cloud program.

References

1. Adelani, D.I., Masiak, M., i, I., Abdulganiyu, H., Omotayo, A.H., Adeeko, A., Afolabi, A., Aremu, A., Samuel, O., Azime, C., Alabi, J., Tonja, A.L., Mwase, C., Ogundepo, O., Dossou, B.F.P., Oladipo, A., Nixdorf, D., Emezue, C.C., sana al azzawi, Sibanda, B., David, D., Ndolela, L., Mukiibi, J., Ajayi, T., Moteu, T., Odhiambo, B., Owodunni, A., Obiefuna, N., Mohamed, M., Muhammad, S.H., Ababu, T.M., Salahudeen, S.A., Yigezu, M.G., Gwadabe, T., Abdulmumin, I., Taye, M., Awoyomi, O., Shode, I., Siro, T.A., Kimotho, W., Ogbu, O., Mbonu, C., Chukwuneke, C., Fanijo, S., Ojo, J., Awosan, O., Kebede, T., Sakayo, T.S., Nyatsine, P., Sidume, F., Yousuf, O., Oduwole, M., Tshinu, T., Kimanuka, U., Diko, T., Nxakama, S., Nigusse, S., Johar, A., Mohamed, S., Hassan, F.M., Mehamed, M.A., Ngabire, E., Jules, J., Ssenkungu, I., Stenetorp, P.: Masakhanews: News topic classification for african languages (2023), https://arxiv.org/abs/2304.09972
2. Alabi, J.O., Adelani, D.I., Mosbach, M., Klakow, D.: Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 4336–4349. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (Oct 2022), https://aclanthology.org/2022.coling-1.382
3. Clark, K., Luong, M., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia. OpenReview.net (2020)
4. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8440–8451. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.747, https://aclanthology.org/2020.acl-main.747

5. Deshpande, V., Pechi, D., Thatte, S., Lialin, V., Rumshisky, A.: Honey, I shrunk the language: Language model behavior at reduced scale. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics: ACL 2023. pp. 5298–5314. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.findings-acl.326, https://aclanthology.org/2023.findings-acl.326
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
7. Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N.: Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping (2020)
8. Finkenstaedt, T., Wolff, D.: Ordered Profusion; Studies in Dictionaries and the English Lexicon. Annales Universitatis Saraviensis, Winter Heidelberg (1973), https://books.google.co.za/books?id=PK-0AAAAIAAJ
9. He, P., Liu, X., Gao, J., Chen, W.: DeBERTa: Decoding-enhanced BERT with disentangled attention. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=XPZIaotutsD
10. Hernandez, D., Brown, T., Conerly, T., DasSarma, N., Drain, D., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Henighan, T., Hume, T., Johnston, S., Mann, B., Olah, C., Olsson, C., Amodei, D., Joseph, N., Kaplan, J., McCandlish, S.: Scaling laws and interpretability of learning from repeated data (2022)
11. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de las Casas, D., Hendricks, L.A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J.W., Sifre, L.: An empirical analysis of compute-optimal large language model training. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022)
12. Jiang, Z., Yu, W., Zhou, D., Chen, Y., Feng, J., Yan, S.: Convbert: improving bert with span-based dynamic convolution. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
13. Kaplan, J., McCandlish, S., Henighan, T., Brown, T.B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling laws for neural language models (2020)
14. Kreutzer, J., Caswell, I., Wang, L., Wahab, A., van Esch, D., Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., Setyawan, M., Sarin, S., Samb, S., Sagot, B., Rivera, C., Rios, A., Papadimitriou, I., Osei, S., Suarez, P.O., Orife, I., Ogueji, K., Rubungo, A.N., Nguyen, T.Q., Müller, M., Müller, A., Muhammad, S.H., Muhammad, N., Mnyakeni, A., Mirzakhalov, J., Matangira, T., Leong, C., Lawson, N., Kudugunta, S., Jernite, Y., Jenny, M., Firat, O., Dossou, B.F.P., Dlamini, S., de Silva, N., Çabuk Ballı, S., Biderman, S., Battisti, A., Baruwa, A., Bapna, A., Baljekar, P., Azime, I.A., Awokoya, A., Ataman, D., Ahia, O., Ahia, O., Agrawal, S., Adeyemi, M.: Quality at a glance: An audit of web-crawled multilingual datasets. Transactions of the Association for Computational Linguistics **10**, 50–72 (2022)

15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach **abs/1907.11692** (2019)
16. Micheli, V., d'Hoffschmidt, M., Fleuret, F.: 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). pp. 7853–7858. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.632, https://aclanthology.org/2020.emnlp-main.632
17. Mohseni, M., Tebbifakhr, A.: MorphoBERT: a Persian NER system with BERT and morphological analysis. In: Freihat, A.A., Abbas, M. (eds.) Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019 - Short Papers. pp. 23–30. Association for Computational Linguistics, Trento, Italy (11–12 Sep 2019), https://aclanthology.org/2019.nsurl-1.4
18. Muennighoff, N., Rush, A., Barak, B., Le Scao, T., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., Raffel, C.A.: Scaling data-constrained language models. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 50358–50376. Curran Associates, Inc. (2023)
19. Nzeyimana, A., Niyongabo Rubungo, A.: KinyaBERT: a morphology-aware Kinyarwanda language model. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 5347–5363. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.367, https://aclanthology.org/2022.acl-long.367
20. Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. pp. 9–16. Proceedings of the Workshop on Challenges in the Management of Large Corpora, Leibniz-Institut für Deutsche Sprache, Cardiff, Wales (2019). https://doi.org/10.14618/ids-pub-9021, http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215
21. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training. Tech. rep., OpenAI (2018)
22. Sun, Z., Li, X., Sun, X., Meng, Y., Ao, X., He, Q., Wu, F., Li, J.: ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. vol. 1, pp. 2065–2075. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.161, https://aclanthology.org/2021.acl-long.161
23. Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H.W., Narang, S., Yogatama, D., Vaswani, A., Metzler, D.: Scale efficiently: Insights from pretraining and finetuning transformers. In: International Conference on Learning Representations (2022)
24. Urbizu, G., San Vicente, I., Saralegi, X., Agerri, R., Soroa, A.: Scaling laws for BERT in low-resource settings. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Findings of the Association for Computational Linguistics. pp. 7771–7789. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.findings-acl.492, https://aclanthology.org/2023.findings-acl.492

25. Visser, R., Grobler, T., Dunaiski, M.: Insights into low-resource language modelling: Improving model performances for south african languages. Journal of Universal Computer Science (2024), in press
26. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Linzen, T., Chrupała, G., Alishahi, A. (eds.) Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/W18-5446, https://aclanthology.org/W18-5446
27. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 483–498. Association for Computational Linguistics, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.41, https://aclanthology.org/2021.naacl-main.41
28. Zhang, Y., Warstadt, A., Li, X., Bowman, S.R.: When do you need billions of words of pretraining data? In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. pp. 1112–1125. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.90, https://aclanthology.org/2021.acl-long.90

Appendix

5.1 Fine-tuning Hyperparameters

For fine-tuning, we conducted a grid search over the following learning rate values:

- **Small Models**: {2.5e-5, 5e-5, 1e-4, 2e-4}
- **Base Models**: {5e-5, 1e-4, 2e-4, 4e-4}

We found that base models performed best with a learning rate of 5e-5, while small models achieved the best results with a learning rate of 1e-4. Additionally, we used the same hyperparameters for our 3M model as for small models and for our 40M model as for base models.

Table 1 provides a detailed list of all fine-tuning hyperparameters. The fine-tuning tasks include CoLA (grammatical acceptability), SST-2 (sentiment classification), MRPC (paraphrase identification), STS-B (semantic similarity), and RTE (textual entailment), as summarized in Table 2.

Table 1. Hyperparameter used when fine-tuning.

Hyperparameter	**Value**
Learning Rate	1e-4 (3M-12M), 5e-5 (40M-110M)
Batch Size	32
Warmup steps	10%
Epochs	3 (GLUE, FLUE, KLUE), 20 (MasakhaNEWS)
Experiment repetitions	3

Table 2. Summary of fine-tuning tasks.

Dataset	**Description**	**Example**
CoLA	Is the sentence grammatical or ungrammatical?	The book was written by John. **(grammatical)**
SST-2	Is the movie review positive or negative?	This movie doesn't care about cleverness, wit or any other kind of intelligent humor. **(negative)**
MRPC	Is sentence A a paraphrase of sentence B?	A) The identical rovers will act as robotic geologists, searching for evidence of past water. B) The rovers act as robotic geologists, moving on six wheels. **(not equivalent)**
STS-B	How similar are sentences A and sentence B from 0 to 5?	A) A man is cutting a potato. B) A woman is cutting a tomato. **(1.25)**
RTE	Does sentence A entail sentence B?	A) Sida does not take any new decisions on support to projects in the Baltic states. B) Baltic Countries join the EU. **(not entailment)**

Table 3. Hyperparameters used when pretraining our various models. We train both ELECTRA and BERT models using small (12M) and base (110M) hyperparameters, while the 3M and 40M parameter configurations only correspond with BERT models.

	Model Size (Parameters)			
Hyperparameter	**3M**	**12M**	**40M**	**110M**
Number of layers	12	12	12	12
Hidden Size	64	256	256	768
FFN inner hidden size	256	1024	1024	3072
Attention heads	1	4	4	12
Attention head size	64	64	64	64
Generator Size (ELECTRA)	N/A	1/4	N/A	1/3
Embedding Size	128	128	768	768
Learning Rate Decay	Linear	Linear	Linear	Linear
Warmup steps	10000	10000	10000	10000
Learning Rate	5e-4	5e-4	2e-4	2e-4
Batch Size	128	128	256	256
Pretraining Steps (BERT/ELECTRA)	1M	1M/1.5M	766K	1M/766K

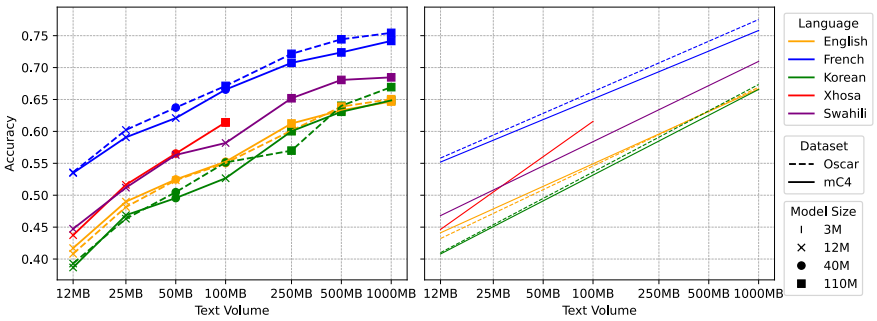

Fig. 6. Optimal BERT model results with increasing pretraining text volumes on the left and the corresponding fitted logarithmic curves on the right. The line for Xhosa is incomplete as the mC4 dataset volume of Xhosa text is less than 250MB.

Table 4. Scaling results for each pretraining dataset of the parameters a and b for the logarithmic fit $y = a\log(x) + b$ along with the R-squared values.

Dataset	a	b	R-squared
English mC4	0.0510	0.3145	0.9642
English Oscar	0.0536	0.2989	0.9706
French mC4	0.0466	0.4361	0.9713
French Oscar	0.0491	0.4363	0.9645
Korean mC4	0.0583	0.2631	0.9747
Korean Oscar	0.0596	0.2619	0.9784
Swahili mC4	0.0546	0.3322	0.9634
Xhosa mC4	0.0798	0.2481	0.9919

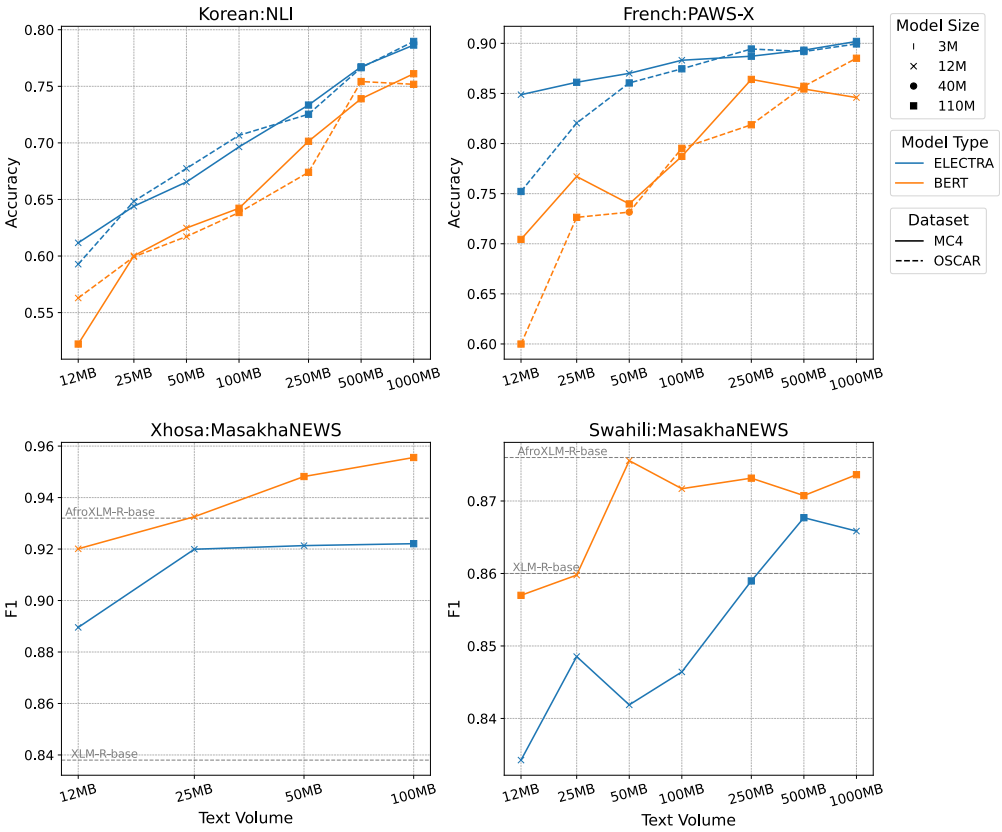

Fig. 7. Downstream performance of optimal BERT and ELECTRA models trained using varying pretraining text volumes. The different line colors indicate which pretraining objective was used, while the line style corresponds with the common crawl dataset sampled for the pretraining text. Each subplot indicates the optimal model performances for a specific downstream task, evaluated using the appropriate task metric.

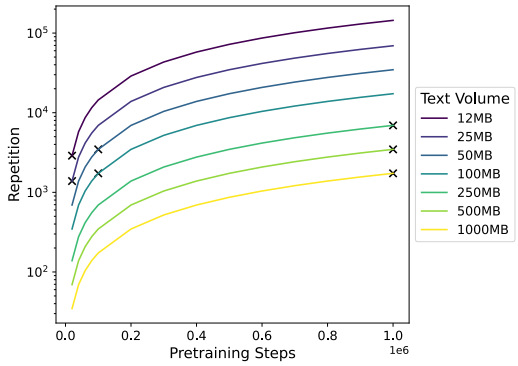

Fig. 8. Number of times input data is repeated during training of BERT-base models. The markers indicate the pretraining step counts at which optimal model performance is observed when sampled from the English Oscar dataset.

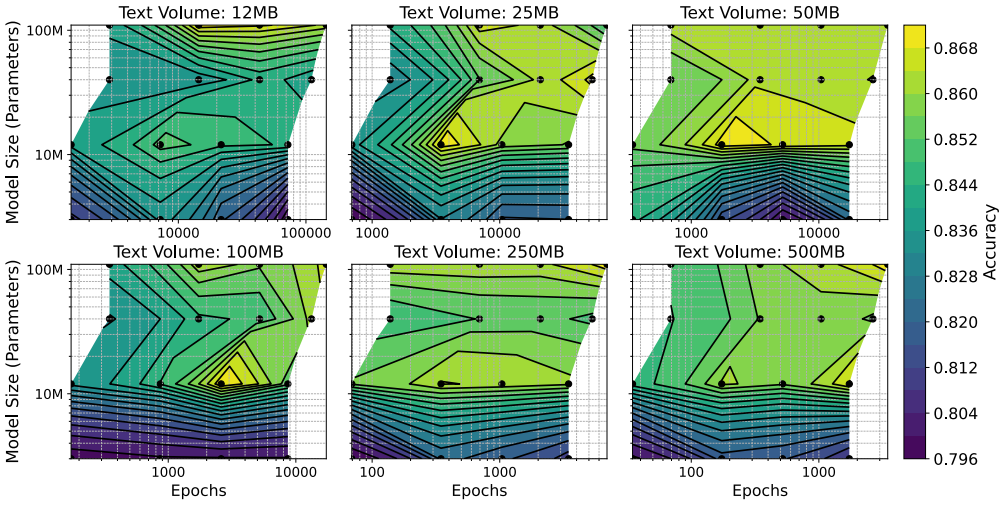

Fig. 9. Contour plots of the MLM accuracy of models using varying amounts of epochs and parameters. Each subplot shows the performance of 16 model checkpoints trained with specific data volumes of Swahili mC4 text.