Fine tuning BLOOM 3B for prompt completion in Zulu.

1st Wisdom Aduah
AI for Science
African Institute for Mathematical Sciences
Muizenberg, South Africa
wizzy@aims.ac.za

***Abstract*—I fine tuned the BLOOM 3B large language model on the Zulu language from the aya dataset from Huggingface using Low Rank Adaptation and Prefix tuning. The models were evaluated on an evaluation dataset, giving chrF++ and Bert scores of 28.83 and 0.86 respectively for LoRA.**

***Index Terms*—LoRA, Fine tuning, LLMs, BLOOM 3B, PEFT**

I. Introduction

A. Backgound

Large language models are transformer [1] models trained to model and/or generate text by drawing from a probability distribution, one token at a time, given some input text. However, they are trained on huge amounts of data, and encode an equally huge number of parameters, making them less suitable for specialized tasks such as Question and Answering. Due to the scarcity of datasets for some languages, the distribution of languages in the training datasets of these models is not even – most African languages are less represented in these datasets. The BLOOM 560M language model for example is trained with less than 0.03% of the datasets representing African languages. Since training a large language model from scratch for low resource languages is not feasible, we instead could fine-tune a pretrained language model.

B. Objectives

I aim to use Low Rank Adaptation (LoRA), and prefix tuning to fine tune the BLOOM 3B large language model for the task of prompt completion in the Zulu language. In addition, I also seek to measure the performance score of the fine-tuned model.

II. Methodology

A. BLOOM 3B

BLOOM 3B is a multilingual model that belongs to a family of models created to enable public research on large language models (LLMs). It is therefore suitable for downstream tasks such as Information extraction, Question Answering and Summarization.

Training data for this LLM is from 45 natural languages, and 12 programming languages – totaling to 1.5TB of preprocessed text, converted into 350B unique tokens. The pie chart shows the distribution of languages in the training dataset.

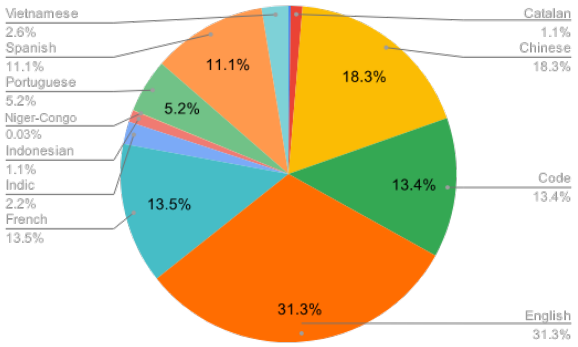

Fig. 1. Distribution of training dataset for the BLOOM 3B.

B. Downstream task

The task is to fine-tune BLOOM 3B for prompt completion using two Parameter Efficient Fine Tuning (PEFT) techniques – Prefix tuning and Low Rank Adaptation (LoRA). To accomplish this, I used the Aya dataset. This dataset is a multilingual instruction fine-tuning dataset containing a total of 204k human-annotated prompt completion pairs. The size of the training dataset is 202k, out of which 1833 annotation pairs are in Zulu language. The same dataset family contains an evaluation suit where there are 200 examples in Zulu.

C. Preprocessing and training procedure

A total of 1833 training examples were used for training. For each training example, a prompt was created by concatenating the input with the target. In addition, each prompt was formatted as follows: ###INPUT input ###TARGET target to make it easier for the model to understand the prompts. For example, after fine tuning, we could simply input "###INPUT input ###TARGET" during prompting and the model would simply complete the prompt. The formatted prompts are then tokenized by the BLOOM tokenizer into input ids, token types, and attention masks. This final form of the processed data was then fed into the transformer for training. The following hyperparameters were used for training: [To be completed latter].

D. Fine-tuning process

LoRA [2] is a PEFT technique which simply decomposes the weight matrix of a specified module, into two matrices. The resulting matrices represent a low rank of the original weight matrix which is then adapted (trained) to specific downstream tasks, reducing the computation complexity and memory footprint. This is very useful especially for our task of fine-tuning the LLM on a low-resource language as Zulu. For our task, our configuration of LoRA resulted in a reduction of the model's trainable parameters from 3022218240 to just 19891200 (0.65%).

Once the training was complete, the model was prompted with the dataset from the Aya evaluation suite, which contains 200 examples. The output of the model was then decoded and compared to the actual targets. To measure this comparison, I used the chrF++ and BERT scores.

III. Experimental setup

I used the following major Python libraries: transformers library from Hugging Face; numpy for numerical computing; PEFT for applying LoRA and Prefix tuning; datasets to get the datasets; Pytorch for tensor computations; matplotlib for visualization. All code was run on Jupyter notebooks hosted on Google colab. Training and inference was done using T4 GPUs provided by colab with the following: RAM: 14.6 GB GPU RAM: 15 GB.

IV. Results

	LoRA	Prefix tuning
chrF++	28.83	22.49
BERt	0.86	0.83

TABLE I
EVALUATION METRICS.

V. Discussion

The model didn't actually do well on the evaluation set as evidenced by the low chrF++ score. I show in the colab notebook some of the generated responses of the model to the inputs of the evaluation dataset. Using google translate to translate the output, it is noticed that most of the responses were completely out of context. For example, the first input in the evaluation dataset is the question "What are 5 ways to eat apples?". One of the responses from the model was something like "Shake it up. Khoziya Mfupa all day long just for the rain. He is not a noble because Mfupa all day long, umzalelo says, and he is punished.". Since Zulu has a very low representation (0.001%) in the training dataset of BLOOM 3B, in addition to the fact that the dataset for fine-tuning was also very little (1833), this result was not surprising.

Fine-tune the model on the English language dataset resulted in generated responses that were much more coherent, and with chrF++ scores as high as 74.

VI. Conclusion

There is a wide performance disparity between high-resource languages and their low-resource counterparts. To bridge this gap, a handful of language models have been built with African languages in mind [3]–[5], resulting in improved performance across several tasks. In future work, I aim to experiment on these Afrocentric models as well.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[3] J. O. Alabi, D. I. Adelani, M. Mosbach, and D. Klakow, "Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning," in *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 4336–4349. [Online]. Available: https://aclanthology.org/2022.coling-1.382

[4] B. F. P. Dossou, A. L. Tonja, O. Yousuf, S. Osei, A. Oppong, I. Shode, O. O. Awoyomi, and C. Emezue, "AfroLM: A self-active learning-based multilingual pretrained language model for 23 African languages," in *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 52–64. [Online]. Available: https://aclanthology.org/2022.sustainlp-1.11

[5] K. Ogueji, Y. Zhu, and J. Lin, "Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages," in *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 116–126. [Online]. Available: https://aclanthology.org/2021.mrl-1.11

.