TextPruner for Low-Resource Language

Pooja Patil
UC, Riverside
ppati010@ucr.edu

Bowen Yang
UC, Riverside
byang095@ucr.edu

Yue Dong
UC, Riverside
yue.dong@ucr.edu

Abstract

This proposed study aims to investigate the performance of XLMR with text pruner before and after pruning for cross-lingual natural language inference (NLI) on the XNLI dataset. Specifically, the study will focus on training with high-resource languages and testing zero-shot as well as few shot performance on low-resource languages. The main objective of this study is to test the hypothesis that TextPruner can retain neurons that deal with cross-lingual understanding abilities for low resource languages. The baseline model is XLMR, which is currently the state-of-the-art model for cross-lingual NLI. The study will evaluate the performance of the proposed approach on the XNLI dataset and compare it with the baseline unpruned model. The results of this study will provide insights into the effectiveness of text pruner for cross-lingual NLI in low-resource settings and its potential for improving the performance of XLMR on this task in compute-restricted environments.

1 Introduction

The advancement in natural language processing, with a heavy reliance on data-hungry large language models and deep neural networks, poses a problem for low-resource languages that lack the large-scale and high-quality datasets required for effective processing (Hedderich et al., 2020) . In this case, cross-lingual models can be leveraged to use pre-trained models in high-resource languages and perform the task in the target language using cross-lingual transfer. (Conneau et al., 2019)XLMR, which is the state-of-the-art model in this domain, has shown promising results for zero-shot cross-lingual transfer for text classification in low-resource languages. However, these larger networks are inaccessible in compute-restricted environments, and this coupled with the low-resource setup of the target language makes the task even more challenging. This is a common situation in NLP for low-resource languages and is referred to as the "low-resource double bind" by (Ahia et al., 2021). As suggested in this paper, pruned models can be helpful in overcoming the "low-resource double-bind" problem.

However, traditional pruning methods often fail to retain the critical neurons responsible for cross-lingual understanding, which leads to a degradation in performance when the pruned models are applied to languages other than the ones they were trained on. TextPruner is an open-source model pruning toolkit that offers fast and easy model compression without the need to retrain the model. Although TextPruner was not specifically designed for cross-lingual understanding tasks, it has shown promising results for retaining neurons responsible for the cross-lingual understanding of the Chinese language, as demonstrated in the (Yang et al., 2022). This problem is particularly acute in low-resource language settings, where the double bind of limited data and cross-lingual complexity can make it difficult to build effective models. The pruning with importance score methodology adopted in TextPruner could still be capable of retaining these critical neurons.

Therefore, this study aims to provide valuable insights into the effectiveness of TextPruner for cross-lingual NLU tasks particularly NLI in low-resource language settings. We seek to shed light on the trade-off between model size and performance and to provide a framework for the development of more efficient and accessible cross-lingual NLU models for low-resource language settings using TextPruner. The importance of evaluating TextPruner's effectiveness in low-resource language settings lies in its potential to provide a solution to the challenges of building effective models in these settings. By evaluating the effectiveness of TextPruner in this context, we can contribute to the existing literature on pruning techniques for cross-lingual understanding in low-resource set-

tings and provide insights into how to build more efficient and effective models for text classification tasks in these settings. Moreover, with TextPruner's easy-to-use interface users with little model training experience can also get access to these effective models.

To keep our research in line with the authors of TextPruner, we begin by reproducing the zero-shot performance results of the pruned XLMR model in English and Chinese languages. Then, we evaluate the same for Swahili, a low-resourced language. We further investigate whether the few-shot condition, in place of the zero-shot condition, makes any difference in the accuracy drops for the pruned models in low-resource language settings.

This report is structured as follows: Section 2 provides an overview of the related work in the field of cross-lingual natural language understanding in low-resource language settings and pruning techniques for language models implemented in TextPruner. Section 3 outlines the methodology used in the project, including the experimental setup and evaluation metrics. Section 4 presents the results of the project. Section 5 discusses the findings and their implications for the development of more efficient and accessible cross-lingual NLU models. Finally, Section 6 concludes the report by summarizing the key findings and their significance for the field. Overall, this project is expected to provide valuable insights into the effectiveness of TextPruner for cross-lingual NLU tasks in low-resource language settings and to contribute to the development of more efficient and accessible language models.

2 Related Work

The main motivation for this work is the unavailability of labeled data for low-resource languages and their failure to leverage the power of large language models thereof. In our work, we will use XLM-R (base)(Conneau et al., 2019) as our base model which will further be pruned and evaluated. XLM-R achieves state-of-the-art performance for cross-lingual understanding and performs particularly well on low-resource languages. It has shown an improvement of 15.7% in XNLI accuracy for Swahili over other XLM models. This makes it the ideal choice for evaluation as our study would make this state-of-the-art performance model accessible in compute-restricted environments. We refer to (Hedderich et al., 2020) who have documented the different dimensions of data availability, the methods that enable training in sparse data situations, and also outline the underlying assumptions of every method for the low-resource setup. It was also pointed out in this survey that low-to-medium depth transformer sizes are more suitable for low-resource languages if we consider monolingual models Hedderich et al., 2020, p.7. It has also been pointed out that few-shot performance is better suited for the low-resource language setup. Now the question arises what performs better a smaller monolingual model or a larger cross-lingual model trained in high-resource language data with fine-tuning in the target language and task? Ahia et al., 2021, p.9 points out that pruning achieves better compression for sparse models and performs better in comparison with the smaller model substitutes. Ma et al., 2021, p.9 also suggests that pruning transformer-based models during training for supervised cross-lingual tasks makes them more applicable to resource-poor languages. Even though these studies focus on pruning during the training phase it sure does open paths to wonder whether pruned pre-trained multi-lingual models would still be able to perform with the same efficiency or if there would be a drop in the performance and how significant it is for cross-lingual tasks. To answer this question our study aims to evaluate a pre-trained model of the XLMR model which was only trained on the English development set after pruning it using an open-source library provided by (Yang et al., 2022). TextPruner offers a comprehensive solution to pruning by reducing the size of multilingual pre-trained language models through structured pruning. Unlike unstructured pruning, which removes individual parameters based on a threshold, structured pruning removes entire rows or columns in weight matrices for faster inference on standard CPU and GPU devices. Additionally, TextPruner simplifies the pruning process for non-expert users and offers pipeline mode for automatic vocabulary and transformer pruning. The major reason for using TextPruner for our case study is that it has shown promising results in retaining neurons responsible for cross-lingual understanding. It was evaluated for zero-shot performance in the Chinese language after pretraining on an English training set and further pruned to 50% of its initial size using an only English development set and yet the drops in the accuracies of the pruned model for Chinese were not greater English, indicating that it

was successful in retaining neurons responsible for cross-lingual understanding. However, since Chinese is considered to be a high-resource language it brings up the question of whether the same hypothesis holds for low-resource languages. Lauscher et al., 2020, p.6 points out that the zero-shot performance of massively multilingual transformers like XLM-R for higher-level cross-lingual understanding tasks drops and shows that there is a correlation between the transfer performance and the size of the pretraining corpora of the target language. Our study aims to extend the evaluation to a low-resource language, Swahili. If TextPruner succeeds to work for low-resource language setups as well then it can be used as a tool for easily making the state-of-the-art models accessible for low-resource languages in compute-restricted environments. To keep our research in line with the previous work we begin with the zero-shot performance of Swahili. (Lauscher et al., 2020) also points out that XLMR has a better performance with a few target language examples. Also, Rosa et al., 2021, p.9 demonstrates that performance on cross-lingual natural language inference tasks improves with original and translated data. This paper also points out that translating datasets for natural language inference tasks is cheap and easy and hence is favorable for low-resource language setup. Inspired by these works, we also further investigate a similar few-shot approach for pruning by providing the pruner with a few target language and task examples.

3 Methodology

In order to evaluate the ability of TextPruner to retain the neurons responsible for cross-lingual understanding our analysis will focus only on the natural language inference tasks. Also, we will be focusing on evaluating the performance of pruning a pre-trained model.

3.1 Dataset

XNLI()(REF) is a benchmark dataset for evaluating cross-lingual understanding tasks. It is created by extending the development and test sets of the MNLI dataset to 15 languages, one of which is our low-resource language Swahili. We will be using accuracy as the evaluation metric for our analysis of the models. The splits for train, development, and testing are kept the same as provided in the dataset. Only minor preprocessing for converting the labels so that they are compatible with our model architecture is carried out. The development sets consist of 2490 examples for every language and the test sets comprise 5010 examples for each language. We have also tried to sample the train sets to get a slightly larger development set to test if more data helps with better importance scores. A simple python script takes in the training data file and splits the data sequentially into some 40 files of around 3000 samples each.

3.2 Model

We continue our experiments on the same model that the authors of TextPruner(Yang et al., 2022) used for demonstrating the cross-lingual understanding retention capability. The model of XLM-ROBERTa(Conneau et al., 2019) was chosen as the baseline model. The model is base-sized with 12 transformer layers with FFN size 3072, hidden size 768, and 12 attention heads per layer. We have used the pre-trained model fine-tuned in the English language for XNLI by the authors of the paper provided on the Hugging face link provided in the repository. We haven't fine-tuned the model in Swahili for the XNLI task but the base model of XLM-R is already pre-trained in 100 languages, including Swahili.

3.3 Pruning Techniques

The advantages of using TextPruner for pruning are that it provides a one-stop solution for pruning both vocabulary and transformer, therefore, providing a great reduction in the size of multilingual pre-trained language models. Traditional pruning methods can be classified into structured and unstructured pruning. Unstructured pruning focuses on individual parameters' magnitude or importance scores and removes them based on some threshold, resulting in sparse matrices. This method provides an inference speed-up only in hardware specialized for dealing with sparse computations. Whereas in structured pruning entire rows or columns in the weight matrices are dropped therefore speeding up the inference process in common CPU and GPU devices. For this reason, TextPruner chooses to implement a structured pruning methodology using the importance scores of the intermediate neurons in the FFN layer or attention heads. These importance scores are calculated by providing a development set to the pruner which it uses to calculate the score and accordingly prune the less important neurons iteratively. The library also allows the user to provide an importance mask which can directly

be used to prune the model in case the user has prior knowledge of the importance scores. For vocabulary pruning, it uses the training/development sets provided by the user and decides whether or not to store the token based on its occurrence in the provided sets. The library also provides a pipeline mode in which both vocabulary and transformer pruning is performed automatically to reduce the full model size. Another advantage of TextPruner is that the easy-to-use interface makes the pruning process accessible to users who are unaware of the model training process details. For our experiments we will be focusing on the iterative method of pruning which calculates the importance scores and then ranks the attention heads and/or neurons in the FFN layers. A lower score means loss is less sensitive to that component and then they're pruned in the order of increasing scores.

3.4 Experimental Design

For our zero-shot experiments, the development set consists of only the English development set and the evaluation will be done on the test splits of English, Chinese, and Swahili. For the few-shot approach, we will use the English development set along with Chinese and Swahili development sets separately as well as both of them combined. This will ensure that the pruner sees the examples for the other languages as well and hopefully be able to estimate the neurons that don't specialize in a single language. Every experiment is carried out for 16 combinations of the number of attention heads per layer (H) and the FFN hidden size per layer (F), which will be denoted as (H, F) for simplicity. The only hyperparameter in our analysis was the number of iterations. As reported by the authors of TextPruner n=16 gives the best results for the experiments. This observation was reverified as well and a similar observation was noted hence this value was chosen for all the other experiments.

4 Experiment

We conducted experiments to evaluate the ability of TextPruner to retain the neurons responsible for cross-lingual understanding in a low-resource language setup, specifically for Swahili using the XNLI dataset. We compared the in-language and zero-shot performance of TextPruner on English and Chinese with the results reported in the original TextPruner paper. We then evaluated the zero-shot performance of TextPruner on Swahili.

4.1 Experimental Setup

We conducted all our experiments on a server with an NVIDIA Tesla P100-PCIE GPU and CUDA version 11.2. The GPU has a memory capacity of 16GB (16280MiB). We used PyTorch version 1.12.1 and the Hugging Face Transformers library version 4.24.0. We used a batch size of 32 and 16 iterations for pruning for all the experiments discussed below. We used the same model and pruning strategy as described in the methodology section for all experiments. All of the experiments are focused on the transformer pruning mode of TextPruner and for homogeneous structure i.e each transformer in the model has the same number of attention heads and same FFN size.

4.2 Evaluation Metrics

We evaluated the models using accuracy, which is a common evaluation metric for natural language inference tasks. The accuracy is calculated as the percentage of correctly predicted labels in the test set.

4.3 Development Set of XNLI

For all the experiments in this section the development sets provided by XNLI are used for computing the importance scores while pruning. The test sets are same as those provided in the dataset.

4.3.1 Reproduction of Previous Results

For in-language analysis the English development set is provided for computing importance scores and English test split is used for evaluating model after pruning. For zero-shot analysis the English development set is provided for computing importance scores and Chinese test split is used for evaluating model after pruning.

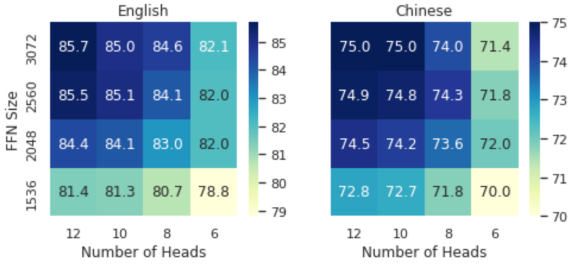

Figure 1: Reproduced results. The results for English are in-language while those for Chinese are zero-shot.

4.3.2 Zero Shot Analysis of Swahili

For zero-shot analysis the English development set is provided for computing importance scores and

Swahili test split is used for evaluating model after pruning.

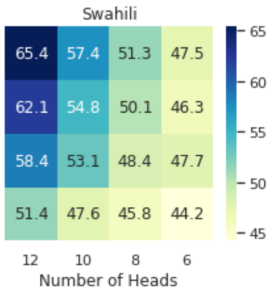

Figure 2: Zero-shot Performance on low-resource language, Swahili

4.3.3 Few Shot Analysis of Chinese

For few-shot analysis the English development set along with Chinese development set is provided for computing importance scores and Chinese test split is used for evaluating model after pruning.

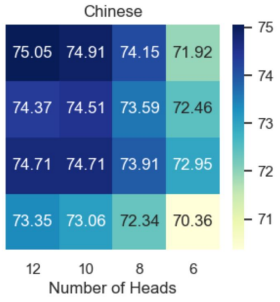

Figure 3: Few Shot Performance on high-resource language, Chinese

4.3.4 Few Shot Analysis of Swahili

For few-shot analysis the English development set along with Swahili development set is provided for computing importance scores and Swahili test split is used for evaluating model after pruning.

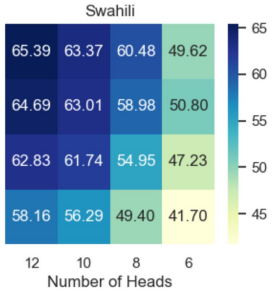

Figure 4: Few Shot Performance on low-resource language, Swahili

4.4 Enhanced Development Set of XNLI

For all the experiments in this section, the development sets were created from the training sets provided by XNLI as described in the Dataset section above. They are used for computing the importance scores while pruning. The test sets are the same as those provided in the dataset. The results for these experiments are averaged over 5 runs to eliminate the factor of chance.

4.4.1 In-language for English and Zero-Shot Analysis of Chinese and Swahili

For this experiment, the English enhanced development set is provided for computing importance scores and English, Chinese, and Swahili test split is used for evaluating model after pruning.

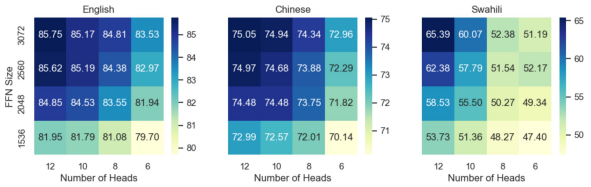

Figure 5: Results for enhanced development set. The results for English are in-language while those for Chinese and Swahili are zero-shot.

4.4.2 In-language for English and Few Shot Analysis of Swahili and Zero-Shot of Chinese

For this experiment, the English enhanced development set is provided along with Swahili enhanced development set for computing importance scores and English, Chinese, and Swahili test split is used for evaluating model after pruning. Note that the performance on Chinese is thus still zero shot.

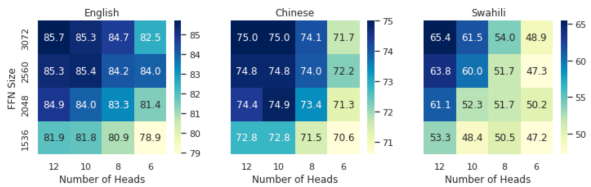

Figure 6: Results for enhanced development set. The results for English are in-language while those for Chinese and Swahili are zero-shot.

4.5 Analysis of Results

We conducted a series of experiments to evaluate the ability of Textpruner to retain the neurons responsible for cross-lingual understanding in low-resource language settings, specifically for Swahili.

Our hypothesis was that Textpruner would be effective in retaining the relevant neurons for cross-lingual understanding even in a low-resource language setting.

To establish a baseline we first replicated the previously published results on the in-language accuracy for English and zero-shot accuracy for high-resource language Chinese using the same experimental setup and parameters as the authors of Textpruner. For the pruned model with FFN size of 1536 and 6 attention heads our results showed a drop in accuracy of 6.9% for English and 5.0% for Chinese after applying Textpruner, which did not exactly match with the results reported in the original paper but the differences were minor and proven statistically insignificant after a pairwise t-test was performed. We then applied Textpruner on the low-resource language Swahili and observed a drop in accuracy of 21.2%. This drop in accuracy was larger than that in Chinese and English, raising a question about whether the hypothesis holds for low-resource language settings. The drop in accuracy seems to be too big for it to hold. We then performed the few-shot analysis by providing examples from Chinese development set and Swahili development set separately. The few-shot analysis results were compared to the zero-shot analysis results and they seem to quite an improvement for Swahili. When a pairwise t-test was carried out it showed statistically significant difference in them. This proves our second hypothesis that the introduction of a few examples from target language can help improve the selection of neurons to be pruned and hence improve the performance. However the accuracy at (1536,6) still seems to fall. The reason behind this observation needs further investigation, one of the reasons for it could be that the structure is just too small to be able to comprise all the neurons responsible for cross-lingual understanding. However if you look at the heatmap in the figure 4 and compare it to the figure 2, you can easily note that for all the other combinations there has been significant improvement. Also, the impact of few shot examples is more apparent in case of low-resource languages compared to the high- resource languages as can be seen from the heatmaps for Chinese and Swahili for zero-shot and few-shot.(refer to figures 1,2,3,4) We also tried to check if more development data positively impacts the pruning performance, and in the zero-shot performance itself we can note that the accuracy for (1536,6) combination has increased by 3.2%, thereby proving that we can improve the performance of pruning by giving more examples in the development set. The authors of the paper tried to evaluate how much data was required to be provided for importance score calculation but they tried to vary the amount from 10-100% of the development set. They failed to evaluate the effect of what happens when we increase them. We can see from figures 5 and 6 that if the development training examples are increased then zero-shot is as good as few-shot. The reason for this is not entirely investigated yet and needs further research to make a conclusive statement.

5 Conclusion

Overall, our experimental results show that Textpruner is indeed effective in retaining the relevant neurons for cross-lingual understanding, but its effectiveness may vary depending on the language and the availability of resources. In order to achieve better performance post-pruning we need to provide the pruner with a few examples from the target low-resource language. Better performance can be observed with a bigger development set as well. Few-shot analysis significantly helps in the case of low-resource language setup. Thus we conclude that if one wishes to use TextPruner for pruning in a low-resource language set up in order to obtain efficient results one should consider few-shot pruning of cross-lingual models for cross-lingual natural language inference tasks. Another alternative one could also try is to provide the pruner with a bigger development set which helps increase the overall performance. In case of scarcity of data alternative methods such as auxiliary data or data augmentation processes can be considered to obtain optimum results. Both enhanced data with few-shot don't show promising results in the preliminary results and need to be further investigated. In conclusion to improve low-resource language performance consider few-shot pruning approach. In order to obtain overall better performance after pruning consider bigger development sets. The current setup can be considered compatible for low-resource language setup with few-shot pruning.

Future work in this field can comprise of suggesting better pruning techniques capable of capturing the cross-lingual understanding capability with fewer data examples so that it can be used for

truly low-resource language setups efficiently. In addition, some data augmentation techinque, like adding some noises

References

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

Weicheng Ma, Kai Zhang, Renze Lou, Lili Wang, and Soroush Vosoughi. 2021. Contributions of transformer attention heads in multi-and cross-lingual tasks. *arXiv preprint arXiv:2108.08375*.

Guilherme Moraes Rosa, Luiz Henrique Bonifacio, Leandro Rodrigues de Souza, Roberto Lotufo, and Rodrigo Nogueira. 2021. A cost-benefit analysis of cross-lingual transfer methods. *arXiv preprint arXiv:2105.06813*.

Ziqing Yang, Yiming Cui, and Zhigang Chen. 2022. Textpruner: A model pruning toolkit for pre-trained language models. *arXiv preprint arXiv:2203.15996*.