Information Retrieval Project Proposal Domain Adaptation using Parameter Efficient Fine Tuning (PEFT)

Janneke Nouwen (s1101750), Daan Brugmans (s1080742), Julian Roddeman (s1080491)

I. Introduction

According to this study (?, ?), sparse and dense LLM-based retrieval models like BERT (Bidirectional Encoder Representations) are less capable of making predictions in unseen domains. BERT is a complex model, fine-tuning all 110 million trainable parameters of the model for an unseen domain is highly expensive, and the overcomplexity of fine-tuning on this number of parameters can lead to overfitting (?, ?).

Parameter Efficient Fine Tuning (PEFT) approaches only fine-tune a small, new set of parameters while keeping the original parameters of the models frozen. This enables rapid adaptations of pre-trained language models to varied domains without fine-tuning the original parameters of the model, lowering computational and storage costs significantly. Because only a small number of parameters is used, this method of fine-tuning is less susceptible to overfitting (?, ?). In this research we will compare the capability of plain BERT to classify unseen documents to a fine-tuned approach using Parameter Efficient Fine Tuning (PEFT) on BERT.

We will use the TREC-COVID dataset from the BEIR framework (?, ?, ?) to determine the model's capability to classify documents in an unseen domain. This decision is based on the fact that BERT was pre-trained solely on an unlabeled, plain text corpus (?, ?) and has no exposure to literature pertaining to the COVID-19 virus, given that its training data predates the pandemic. As a result, the specificity and freshness of the COVID-19 topic certainly designates it as an unseen domain for the model, establishing a valid dataset to assess PEFTs potential to boost the performance of classifying documents of an unseen domain.

II. Research Question

How can a Parameter Efficient Fine Tuning (PEFT) method improve the performance of BERT for classifying documents in unseen domains, when we compare it to a plain BERT setup?

III. Resources

Python 3.10 will be used to test plain BERT and BERT fine-tuned with PEFT. To apply different PEFT methods to BERT, the 'adapter-hub' library will be used (?, ?). We will compare the results using the Scikit-learn package's precision, F1-score and recall (?, ?). For data preprocessing, Numpy and Pandas will be used (?, ?, ?).

IV. Experimental Design

Objective: Our objective is to compare the capability of BERT to classify documents in an unseen domain, comparing a non-fine-tuned plain version of BERT with a dense layer, against a fine-tuned approach with PEFT. We will do this in an unseen domain, specifically in a Covid-19 domain. In figure 1, a flowchart shows the experimental setup. **Hypothesis:** We expect that PEFT will enable BERT to effectively classify documents in an unseen (COVID-19) domain with improved performance compared to a plain BERT model. **Independent Variable:** Fine-tuning approach (plain BERT vs. fine-tuned BERT using PEFT) **Dependent Variable:** Model's performance in classifying relevancy labels of COVID-19 documents using the precision, recall and F1-score.

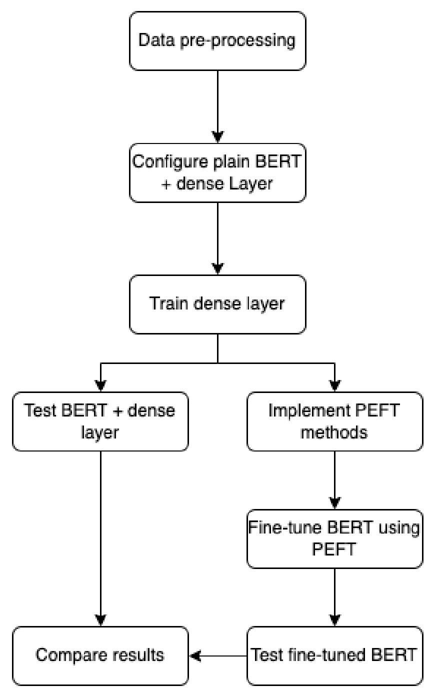

Fig. 1. Flowchart of the experiment setup

References

Bejani, M. M., & Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 1–48.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Vol. 445, pp. 51–56).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., … Gurevych, I. (2020, October). AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 46–54). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-demos.7 doi: 10.18653/v1/2020.emnlp-demos.7

Soekhoe, D., Putten, P., & Plaat, A. (2016, 10). On the impact of data set size in transfer learning using deep neural networks. In (p. 50-60). doi: 10.1007/978-3-319-46349-0_5

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W. R., Lo, K., … Wang, L. L. (2021). Trec-covid: constructing a pandemic information retrieval test collection. In *Acm sigir forum* (Vol. 54, pp. 1–12).