Evaluating Parameter-Efficient Finetuning Approaches for Pre-trained Language Models on the Financial Domain: Replication on Google Colab[*]

Nurseiit Bakkali
MSc Artificial Intelligence
Cardiff University
bakkalin@cardiff.ac.uk

Su Pyae Thu Ya
MSc Natural Language Processing
Cardiff University
yasp@cardiff.ac.uk

Abstract

The rapid rise of large-scale language models introduced challenges like computational cost and storage, especially when fully fine-tuning is required to apply them on the special tasks. Keeping fine-tuned weights for each task is a problem for storage, and the time spent on each fine-tuning is gradual. Parameter-efficient tuning (PEFT) methods offer a promising alternative by freezing the pretrained weights and updating only a small number of additional parameters. Although these methods have shown competitive performance on general NLP benchmarks such as GLUE, their application in the financial sphere is not fully explored. In this paper, we replicate the results of the referenced original paper, where they compares full fine tuning with two PEFT techniques like LoRA and Adapter on financial domain-specific BERT models. Our replication confirms that parameter-efficient approaches can achieve nearly the same results in the financial sphere as full fine-tuning while significantly reducing training time and computational resource usage.

1 Introduction

Natural Language Processing (NLP) and the rise of the large language models transformed the way machines understand human language. This transformation is also impactful in the financial domain, where understanding and processing the text is important. Financial data often has the form of unstructured data such as financial reports, news articles, and investor commentary which can offer valuable insights for tasks like risk analysis or investment decision-making. To use this data researches has adaptad large pre-trained models like BERT[2] for financial application, making the models like FinBERT[18] and FLANG-BERT[12]. These models are powerful, but require fine-tuning on specific tasks which have high cost and storage.

To handle this issue, more efficient fine tuning approaches have emerged such as LoRa[5] and Adapters[4]. These new approaches focus on reducing the number of trainable parameters by freezing original weights and training small additional components. This significantly decreases the computation cost and storage space, while still giving nearly the same performance compared to full fine-tuning. Although these methods showed comparable results to full fine-tuning on GLUE[15], they are less explored in the financial sphere.

In this work, as we are replicating we do the same as in the original paper[7] and evaluate how well this PEFT methods perform on financial tasks using the BERT-like models. Specifically, we (a) compare the performance to fully fine-tuned models across four financial tasks, and (b) highlight their advantage in terms of the smaller number of trainable parameters.

2 Related Work

There has been extensive research on the applying of the PEFT methods on the GLUE benchmark [9][10][3]. In financial domain, number of experiments done by training model from scratch or applying continued pretraining on financial text and comparing these domain-specific models to

[*] https://colab.research.google.com/drive/1Kjzc9WSghctUIXwAOh0g8dLF-UuYrBgh?usp=sharing

general-purpose ones[8][18][12]. These efforts are valuable, but did not use PEFT methods while fine-tuning. Also, there are large language models in the financial domain[16][17], but they are not in open access or don't make use of PEFT methods. However, [18] authors first did a mix of LoRa fine-tuning and reinforcement learning while building the financial domain model. As we are replicating, our paper[7] was the first research in the application of the PEFT methods in the financial sphere using BERT-like models. Along with this, our paper is the replication of this paper and confirmation of results that PEFT mehtods are efficient enough to apply instead of the full fine-tuning.

3 Methodology

In the following discussion, we describe the models employed, the fine-tuning strategies applied, the types of tasks addressed, and the corresponding hyperparameters used for each method.

3.1 Models

Three baseline models are used in this experiment: `BERT-base-uncased`[2], `FinBERT`[18], and `FLANG-BERT` [12], with the latter two specialized for the financial domain

BERT-base: is a general-purpose pretrained language model that utilizes a bidirectional transformer architecture and is trained on a large corpus of English text. It employs WordPiece tokenization and self-supervised objectives, which are masked language modeling and next sentence prediction.

FinBERT: is tailored version of BERT, especially for financial domain, developed to better capture the nuances of financial language. Using the same transformer architecture as BERT, it is pretrained on a large-scale corpus comprising earnings call transcripts, corporate filings, and analyst reports.

FLANG-BERT: is also one of the latest financial language models developed through extension of pretrained BERT on finance-related data, incorporating financial terminology and domain-specific expressions.

3.2 Fine-tuning Techniques

LoRA (Low-Rank Adaptation): is a lightweight fine-tuning method that modifies large pretrained language models by introducing additional low-rank trainable matrices into each layer, while keeping the original model parameters fixed[5]. This approach greatly lowers the number of parameters that need to be updated and reduces computational demands. Despite its efficiency, LoRA maintains performance comparable to that of full model fine-tuning across a range of Transformer architectures.

Adapter$_\mathbf{H}$ (Houlsby Adapter): introduces a parameter-efficient fine-tuning approach by inserting small bottleneck adapter modules within each layer of a pretrained Transformer model [4]. These adapters consist of a down-projection to a lower-dimensional space, a non-linear activation, and an up-projection back to the original dimension, all integrated with residual connections. During fine-tuning, only the adapter parameters are updated while the original model weights remain frozen, enabling efficient adaptation to new tasks with minimal additional parameters.

3.3 Financial Tasks

There are altogether four downstream tasks from the Financial Language Understanding Evaluation (FLUE) benchmark [12]. For each task, the sample dataset is enumerated to the number that the original paper has provided.

Sentiment Classification: The Financial Phrasebank dataset from the hugging face `takala/financial_phrasebank` [6], using the split `sentences_50agree`, where at least 50% of the annotators agreed on the sentence labels, has been used for classifying sentiment. The dataset comprises 4,846 sentences, each labeled as one of three classes: positive (2), neutral (1), or negative (0).

Sentiment Regression: The FiQA dataset files have been downloaded from `dayanfcosta/fiqa-2018-task1` GitHub repository[1]. Two training dataset files, consisting of headlines and posts, were merged and a random sample of 1,173 entries was selected to match the original paper. The dataset contains sentiment scores on a continuous scale from -1 to 1.

News Headline Classification: We used the Gold News Headline dataset[13] to detect time sensitive data in financial news such as price fluctuation. It has 11 412 headlines and nine corresponding tags such as *Past Price, Future News, or Asset Comparison*. The goal is to perform multilabel classification with 9 classes.

Named Entity Recognition: Named Entity Recognition (NER) plays a key role in financial text analysis by identifying important entities such as place, people and organizations which helps to understand the relationship between them. Dataset is taken from hugging face `tner/fin` which is on this [14] paper. The original paper referenced [11] paper, but there is no publicly available data. The dataset which we found is the same as the in referenced paper. The dataset consist of 1466 annotated samples and four entities: PERSON (PER), LOCATION (LOC), ORGANISATION (ORG) and MISCELLANIOUS (MISC). FIN5.txt is for train and validation and FIN3.txt for test data.

3.4 Hyperparameters Selection

Hyperparameter search for the `BERT-base` model has been conducted to determine the optimal learning rate and epochs for each financial task. The learning rate is consistent across all models, varying within the range $\{1e{-}6,\ 1e{-}5,\ 2e{-}5,\ 3e{-}5,\ 5e{-}5,\ 1e{-}4,\ 1e{-}3\}$. As for the number of epochs, $[3,\ 4,\ 5]$ is used for full fine-tuning and LoRA, while $[6,\ 9,\ 11]$ is used for Adapter$_\mathrm{H}$.

Due to technical limitations and time constraints, the same batch size, LoRA dropout rate, rank, and alpha value are implemented from the results reported in the original paper for each of the tasks. However, for some tasks, the provided LoRA parameters did not perform well and were therefore adjusted. Using the same hyperparameters as in the search, the `FinBERT` and `FLANG-BERT` models are then evaluated. The results of the models are recorded in detail in section 4.

4 Findings and Analysis

The replicated results are provided in 4 separate tables, one for each task, along with the original to compare the performance. The models are categorized into three types: fully fine-tuned, fine-tuned with LoRA, and fine-tuned with Adapter$_\mathrm{H}$. This categorization applies to both `FinBERT` and `FLANG-BERT`.

4.1 Financial Phrasebank

Table 1 presents the Financial Phrasebank results. Although the replicated results differ slightly from the original, fine-tuning `BERT-base` achieved the highest performance, with Adapter$_\mathrm{H}$ yielding similar results. For `FinBERT`, the Adapter$_\mathrm{H}$ model outperformed the baseline and that using LoRA. Surprisingly, for `FLANG-BERT`, all three fine-tuning methods produced comparable results, with Adapter$_\mathrm{H}$ achieving the best overall performance.

4.2 FiQA

The results on the FIQA dataset are presented on Table 2. In this case, while the overall trends differ slightly from the Financial Phrasebank, some consistent patterns remain. For `BERT-base`, traditional fine-tuning continued to deliver the best performance. In contrast, for `FinBERT`, the baseline model gained superior performance over both LoRA and Adapter$_\mathrm{H}$. Meanwhile, `FLANG-BERT` showed its best performance when fine-tuned using Adapter$_\mathrm{H}$, outperforming the other two variants, although the fine-tuned baseline model demonstrated comparable results.

4.3 Gold News Headlines

In Table 3 for News Headline Classification results, we can see that FLANG-BERT+Adapter model is the best one compared to other models. In BERT case there is not so much difference between PEFT models and base BERT model. For FinBERT model case using adapters showed best performance. In overall, PEFT models showed approximately the same accuracy as the full fine-tuned BERT-like models. Our results, also little bit differs here, but shows the same trend. The number of trainable parameters in LoRa is 0.41% and in Adapter it is 1.61%. However, Lora model performs nearly same as Adapter one even it has smaller number of paramters.

4.4 NER

Table 4 shows the compared results for NER task. Compared to the original paper we got little bit higher F-1 scores. However, in both results full fine-tuned models are the best performing ones. In BERT and FinBERT cases, adapter models got near F-1 score compared with full fine-tuned model version. In FLANG-BERT case, LoRa model showed nearest F-1 score to the full fine-tuned model. Overall, this task also showed that PEFT method can achieve nearly the same results while maintaing space and computational cost. Lora model has 0.81% of parameters whereas

Adapter has 1.62% paramaters and both performs good. Some major differences of our results will be due to the fact that the dataset which they provided was unaccessible and we find by ourselves the same dataset. This dataset will be updated version of the initial one. However, the size of both dataset is the same. While training the models, we skipped empty batches which do not contained entities and this will also affect to the total score.

All replicated models outperformed the original results, except for `FinBERT` on the Financial Phrasebank dataset.

5 Discussion

In our replication of the original study, we evaluated the performance and effectiveness of the parameter -efficient fine-tuning methods in financial domain by comparing them to the traditional full fine-tuning. Our overall findings support the original paper's conclusion and methods like LoRa and Adapters achieve approxiemately the same results while maintaining the size and decreasing the computational cost. We observed some variation in specific tasks.

In small datasets like FiQA and NER models achieve the same accuracies with smaller number of parameters like reported in original paper. This less number of parameters would be good while training as it helps to avoid overfitting and catastrophic forgetting.

In our replication, we also observed that Adapter models perform better than LoRa methods. This is explained in the original paper by saying that Adapter has the more trainable parameters. The suprising thing was, while training LoRa we found that it gets the same evaluation score on the smaller number of epochs like 3,4 or 5. It was the main difference between our paper and the original one.

Having smaller PEFT parameters allow us to save them locally on smaller space and use them depending on the task. In our experiment we noticed that PEFT methods run faster, but we did not measured the time.

6 Limitations

This experiment encountered several limitations while attempting to replicate the original paper.

Models	**Original**	**Replication**
	Accuracy	Accuracy
BERT-base	0.86	**0.861**
BERT-base + LoRA	0.83	0.842
BERT-base + Adapter$_\text{H}$	0.86	0.858
FinBERT	0.86	0.843
FinBERT + LoRA	0.89	0.814
FinBERT + Adapter$_\text{H}$	0.86	**0.849**
FLANG-BERT	0.86	0.854
FLANG-BERT + LoRA	0.84	0.856
FLANG-BERT + Adapter$_\text{H}$	0.86	**0.857**

Table 1: Financial Phrasebank replication comparison

Models	**Original**	**Replication**
	MSE	MSE
BERT-base	0.12	**0.077**
BERT-base + LoRA	0.16	0.105
BERT-base + Adapter$_\text{H}$	0.10	0.091
FinBERT	0.09	**0.072**
FinBERT + LoRA	0.08	0.92
FinBERT + Adapter$_\text{H}$	0.09	0.104
FLANG-BERT	0.08	0.075
FLANG-BERT + LoRA	0.17	0.078
FLANG-BERT + Adapter$_\text{H}$	0.07	**0.074**

Table 2: FiQA replication comparison

Models	**Original**	**Replication**
	Mean F-1	Mean F-1
BERT-base	0.97	**0.968**
BERT-base + LoRA	0.95	0.961
BERT-base + Adapter$_\text{H}$	**0.97**	0.963
FinBERT	0.97	0.967
FinBERT + LoRA	0.94	0.958
FinBERT + Adapter$_\text{H}$	**0.97**	**0.967**
FLANG-BERT	0.97	0.968
FLANG-BERT + LoRA	0.95	0.965
FLANG-BERT + Adapter$_\text{H}$	**0.97**	**0.969**

Table 3: News Headline Classification replication comparison

Models	**Original**	**Replication**
	F-1	F-1
BERT-base	0.81	**0.811**
BERT-base + LoRA	0.75	0.787
BERT-base + Adapter$_\text{H}$	0.77	0.803
FinBERT	0.77	$\underline{0}$.825
FinBERT + LoRA	0.76	0.778
FinBERT + Adapter$_\text{H}$	0.76	0.793
FLANG-BERT	0.81	**0.819**
FLANG-BERT + LoRA	0.78	0.806
FLANG-BERT + Adapter$_\text{H}$	0.76	0.768

Table 4: NER replication comparison

Models	Trainable Parameters
BERT-base	109,484,547 (100%)
BERT-base + LoRA	887,043 (0.80%)
BERT-base + Adapter$_H$	3,004,605 (2.67%)
FinBERT	109,754,115 (100%)
FinBERT + LoRA	887,043 (0.80%)
FinBERT + Adapter$_H$	3,004,956 (2.66%)
FLANG-BERT	109,484,547 (100%)
FLANG-BERT + LoRA	887,043 (0.80%)
FLANG-BERT + Adapter$_H$	3,004,605 (2.67%)

Table 5: Comparison of trainable parameters for Financial Phrasebank

Models	Trainable Parameters
BERT-base	109,483,009 (100%)
BERT-base + LoRA	885,505 (0.80%)
BERT-base + Adapter$_H$	2,381,953 (2.13%)
FinBERT	109,752,577 (100%)
FinBERT + LoRA	885,505 (0.80%)
FinBERT + Adapter$_H$	2,381,953 (2.12%)
FLANG-BERT	109,483,009 (100%)
FLANG-BERT + LoRA	885,505 (0.80%)
FLANG-BERT + Adapter$_H$	2,381,953 (2.13%)

Table 6: Comparison of trainable parameters for FiQA

Models	Trainable Parameters
BERT-base	109,489,161 (100%)
BERT-base + LoRA	449,289 (0.41%)
BERT-base + Adapter$_H$	1,795,977 (1.61%)
FinBERT	109,758,729 (100%)
FinBERT + LoRA	449,289 (0.41%)
FinBERT + Adapter$_H$	1,795,977 (1.61%)
FLANG-BERT	109,489,161 (100%)
FLANG-BERT + LoRA	449,289 (0.41%)
FLANG-BERT + Adapter$_H$	1,795,977 (1.61%)

Table 7: Comparison of trainable parameters for News Headline classification

Models	Trainable Parameters
BERT-base	108,895,493 (100%)
BERT-base + LoRA	888,581 (0.81%)
BERT-base + Adapter$_H$	1,792,901 (1.62%)
FinBERT	109,165,061 (100%)
FinBERT + LoRA	888,581 (0.81%)
FinBERT + Adapter$_H$	1,792,901 (1.62%)
FLANG-BERT	108,895,493 (100%)
FLANG-BERT + LoRA	888,581 (0.81%)
FLANG-BERT + Adapter$_H$	1,792,901 (1.62%)

Table 8: Comparison of trainable parameters for NER task

Unlike the original study, which utilized Luxembourg's national supercomputer MeluXina, we conducted our experiments on Google Colab Pro using an L4 GPU. This imposed greater constraints on our choice of hyperparameters, including learning rate, number of epochs, batch size, LoRA rank, alpha, and dropout rate. As a result, this led to a reduction in the number of tunable parameters and the use of more static values in some places.

7 Conclusion

In conclusion, we confirm reults of original paper. We showed that parameter-efficient fine-tuning methods can reach the same performance regardless of the number of trainable parameters. Using of this techniques will be more practical in Financial sphere and reduce cost in computational power. Further, like in original paper we suggest using of the other PEFT mehtods to research how they will affect to the model performance. Also, Large language models are good option to try in financial sphere. Applying PEFT methods on them will open new opportunites for practical usage of ML in finance and will beneficial for environment by reducing carbon footprints.

References

[1] Dayan de França Costa and Nádia Felix Felipe da Silva. Inf-ufg at FiQA 2018 task 1: Predicting sentiments and aspects on financial tweets and news headlines. In *Proceedings of the Companion of The Web Conference 2018*, pages 1967–1971. International World Wide Web Conferences Steering Committee, 2018.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. 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 2208–2222, Online, 2021. Association for Computational Linguistics.

[4] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[6] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014.

[7] Isabella Olariu, Cedric Lothritz, Jacques Klein, Tegawendé Bissyandé, Siwen Guo, and Shohreh Haddadan. Evaluating parameter-efficient finetuning approaches for pre-trained models on the financial domain. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15482–15491, Singapore, December 2023. Association for Computational Linguistics.

[8] Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. Is domain adaptation worth your investment? comparing bert and finbert on financial tasks. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 37–44, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[9] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, 2020. Association for Computational Linguistics.

[10] Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[11] Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain adaptation of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia, 2015. Australasian Language Technology Association.

[12] Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. When flue meets flang: Benchmarks and large pretrained language model for financial domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2022.

[13] Ankur Sinha and Tanmay Khandait. Impact of news on the commodity market: Dataset and results. *arXiv preprint arXiv:2009.04202*, 2020.

[14] Asahi Ushio and Jose Camacho-Collados. T-ner: An all-round python library for transformer-based named entity recognition. *arXiv preprint arXiv:2209.12616*, 2022.

[15] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 2018. Association for Computational Linguistics.

[16] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann.

Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[17] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv preprint arXiv:2308.11458*, 2023.

[18] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*, 2020.