Sentiment Analysis on Parameter-Efficient Neural Distillate with Iterative Fine-Tuning (SEND-IT)

Project Team: SEND-IT
J. Henderson, P. Tamilselvan, S. Allauddin, T. Taprantzis

INTRODUCTION

The exponential growth of large language models (LLMs) has revolutionized natural language processing (NLP), enabling state-of-the-art performance across a wide range of tasks. However, the computational and memory demands of full fine-tuning for these models present significant challenges, especially for researchers and practitioners working with limited resources, which necessitates lightweight implementation and experimentation for pupils and independent practitioners alike. As a result, Parameter-Efficient Fine-Tuning (PEFT) methods have emerged as compelling means of facilitating experimentation with larger LLMs, offering reduced computational cost while preserving performance.

This study explores the efficacy of two prominent PEFT techniques—Low-Rank Adaptation (LoRA) and BitFit—for sentiment analysis in low-resource environments. LoRA approximates full weight matrices using low-rank factorization to reduce the number of trainable parameters, while BitFit takes a sparse approach by updating only bias terms. These methods promise to strike a balance between model performance and efficiency, which is critical for real-world deployment on resource-constrained systems.

Through this work, we aim to provide a comparative evaluation of LoRA and BitFit in terms of classification accuracy, memory footprint, and computational overhead. The findings will offer insights into the viability of these fine-tuning strategies in practical, low-resource NLP applications and contribute to broader discussions on scalable and sustainable AI model development.

BACKGROUND

Recent developments in Parameter-Efficient Fine-Tuning (PEFT) have aimed to reduce the cost of adapting large pre-trained language models without significantly sacrificing performance. LoRA and BitFit represent two distinct approaches within this space. LoRA introduces trainable low-rank matrices into each layer, reducing the number of parameters updated during training while preserving expressive capacity. BitFit, in contrast, simplifies fine-tuning by updating only the bias terms, offering a minimal yet surprisingly effective adaptation strategy under constrained conditions [1].

Dataset:

We focus on a binary sentiment classification task using the Sentiment140 dataset, a large-scale corpus of annotated tweets, to benchmark these PEFT techniques [2]. To further simulate real-world constraints, we downsample the dataset and conduct training on consumer-grade hardware. Additionally, we incorporate knowledge distillation, aiming to evaluate the combined impact of distillation and PEFT on model size, accuracy, and runtime.

Related Work:

Rust et al. (2023) highlight the trade-offs between these methods, noting LoRA's superior performance at the cost of higher memory usage and BitFit's efficiency with lower accuracy in high-complexity settings [3]. Other studies, such as Lester et al. (2021), explore alternatives like prompt tuning but emphasize the need for careful prompt design and acknowledge limitations in resource efficiency during training [4].

METHODOLOGY

This project evaluates two Parameter-Efficient Fine-Tuning (PEFT) techniques—Low-Rank Adaptation (LoRA) and BitFit—on a binary sentiment classification task using a compact off-the-shelf language model and a teacher-student distillation thereof. LoRA works by decomposing weight matrices into low-rank representations, enabling updates to a much smaller subset of parameters. We fine-tune models by selecting appropriate rank values and using iterative training to balance accuracy and efficiency. BitFit, in contrast, takes a sparse approach by freezing all model weights and updating only the bias terms. This drastically reduces memory and compute requirements, making it especially suitable for local training environments. Both methods are implemented using HuggingFace Transformers, with experiments conducted on OPT-125M and smaller distilled versions to further reduce the resource load [5].

To further defend the objective of bringing tractable experimentation on larger-scale LLMs (on the order of 125m parameters, such as Meta's OPT model) within reach to the independent practitioner such that they can experiment on consumer-grade hardware, we further constrain our experiments by performing pre-processing on the Sentiment140 dataset. We begin by dividing our dataset into positive and negative classes. For sake of completeness, there was said to have been a neutral class, but no such instances were found (perhaps this is an artifact from a previous dataset version). We perform stratified downsampling to 50k instances (from the original 1.6m tweets) and on the remaining data we apply denoising such as stripping mentions (e.g. @gatech), URLs (e.g. http://…), hashtags (e.g. #SwineFlu) and non-alphanumeric characters such as emojis, capitalization and punctuation. We then subsequently split this smaller dataset 80/10/10 amongst Train/Test/Validation sets. An example of the output, pre-processed data is provided in Fig. 1 directly below.

To ensure fair and meaningful comparisons, we run multiple

Training	text	user	sentiment
47782	finally set up wireless internet huzzah for tw...	alexwilliamson	1
20407	lebron james please dont leave usfor the love ...	ryangetty	0
42997	i broke our site	DjDATZ	0
19678	ugh idk if thats going to be possible my frie...	JulieAnnCook	0
13754	wrong place at the wrong time always sigh	Nadiahazman	0

Fig. 1: Sample of labeled training tweets (pre-processed and denoised)

training trials for each method, using supervised learning with cross-entropy loss. Hyperparameter tuning is performed using grid search over learning rate, batch size, and number of epochs. We evaluate models based on three metrics: classification accuracy on a held-out test set, memory usage during training, and overall runtime. These metrics reflect trade-offs between performance and efficiency, which are critical in low-resource settings. All training and evaluation are carried out on consumer-grade machines (e.g., M2 MacBook Air) without access to GPUs, making practical efficiency a central focus of this study.

RESULTS & ANALYSIS

Baseline Results:

Before experimentation with model distillation, where we establish a teacher-student dynamic between Meta AI's OPT-1.5b and OPT-350m (as teacher) and OPT-125m (as student) to subsequently evaluate as an effective means of overclocking our base OPT-125m model, we first establish the baseline model described below.

The baseline OPT-125m LLM is used to run inference on the Sentiment140 dataset, which we take a 50k sample subset of in the steps outlined in Methodology, and evaluate via performance metrics as follows in Fig. 2.

OPT-125m Inference on Sentiment140 Dataset (Preprocessed, Downsampled to 50k instances)				
Test Accuracy			0.5312	
F1 Score (macro):			0.5305	
F1 Score (weighted):			0.5305	
Inference Time:		[s]	27.51	
Classification Results				
	Precision	Recall	F1-Score	Support
Negative	0.53	0.49	0.51	2500
Positive	0.53	0.57	0.55	2500
Accuracy			0.53	5000
Macro Average	0.53	0.53	0.53	5000
Weighted average	0.53	0.53	0.53	5000

Fig. 2: Summary inference results of pre-PEFT OPT-125m on Sentiment140.

We also visualize performance on inference via a confusion matrix in Fig. 3 at right.

Most notably, from Fig. 2. When we tabulate inference results, we observe that without tuning (particularly the PEFT techniques to come), none of our key metrics across test accuracy, f1 score or its derivative precision and recall are performing much better than random chance for a binary classification task. This is less obviously true in our confusion matrix above in Fig. 3., though visually we can discern that while our model is strongest in picking up on positive sentiment when the tweet is indeed positive (true positives), there is a similarly high rate of false positives and negatives.

We can do better, as will be illustrated in the PEFT baseline (and Model Distillation, with and without PEFT) results sections that follow.

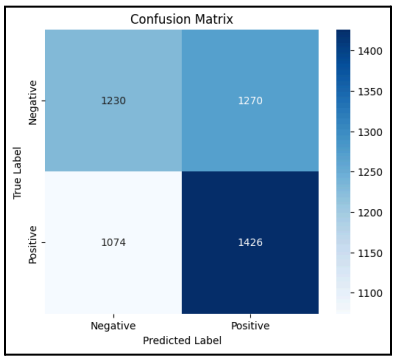

Fig. 3: Confusion matrix of binary accuracy (pre-PEFT OPT-125m model).

PEFT Baseline Results:

We evaluate our two PEFT approaches, LoRA and BitFit, on the binary sentiment classification task on the Sentiment140 dataset. We performed consistent hyperparameter tuning between the two PEFT approaches, namely in scaling learning rate and batch size, and we selected the best based on the same performance metrics as before.

LoRA:

In applying LoRA (Low-Rank Adaptation) to our OPT-125m model, we freeze the baseline model and evaluate performance on the new low-rank adaptations of the baseline.

Due to the runtime-intensive implications of training a dataset of this size with embedded hyperparameter tuning (sweeping learning rate between 5e-5 and 1e-4 and batch size between 8 and 16), we previously ran all training regimes for 6 epochs before noticing a plateau in validation accuracy after 3 epochs, so full-scale training was performed across all non-distilled model experiments with just 3 epochs.

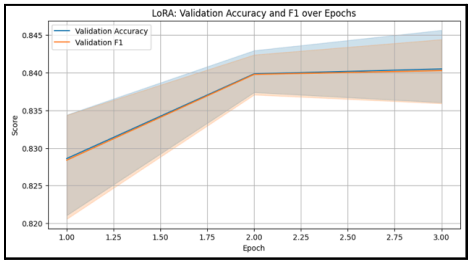

Fig 4: Learning curve of val acc and F1 on baseline OPT-125m w/ LoRA.

The learning curve for LoRA applied to the OPT-125m model trained on the Sentiment140 dataset for the binary sentiment analysis task is given above in Fig. 4., where we observe a drastic improvement in test accuracy (0.84) over the near-random-chance accuracy on inference observed of the pre-PEFT baseline (0.53).

We observe through hyperparameter search, as summarized in Fig. 5. Below, the best tradeoff between validation accuracy,

F1-score, training runtime and memory usage is associated with lr=5e-5 and bs=16, shown in the bottom row. Max GPU usage on model training with LoRA was 5.07027GB.

OPT-125m w/ LoRA - Training Metrics							
	lr	batch_size	best_val_f1	val_accuracy	runtime_sec	memory_delta_mb	Test Accuracy
3	0.0001	16	0.847686	0.8478	1667.428003	0.929688	0.8414
2	0.0001	8	0.843577	0.84	1731.900387	0.644531	
0	0.00005	8	0.839095	0.8392	1742.056741	1.554688	Test F1 (macro)
1	0.00005	16	0.8358	0.835	1668.280001	0.511719	0.8412

Fig 5: Baseline OPT-125m w/ LoRA - summary training performance metrics.

OPT-125m Inference on Sentiment140 Dataset w/ LoRA (Preprocessed, Downsampled to 50k instances)				
Test Accuracy			0.8414	
F1 Score (macro):			0.8412	
F1 Score (weighted):			0.8412	
Inference Time:		[s]	31.11	
Classification Results				
	Precision	Recall	F1-Score	Support
Negative	0.82	0.87	0.85	2500
Positive	0.87	0.81	0.84	2500
Accuracy			0.84	5000
Macro Average	0.84	0.84	0.84	5000
Weighted average	0.84	0.84	0.84	5000

Fig 6: Baseline OPT-125m with LoRA - summary inference results.

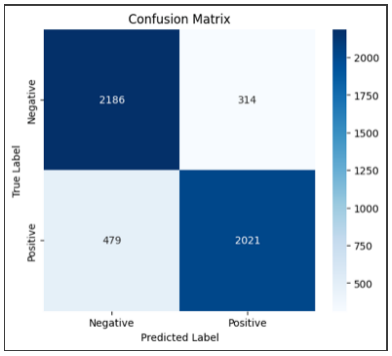

Fig 7: Confusion matrix of binary accuracy, baseline OPT-125m w/ LoRA.

We also observe on inference (e.g. Fig. 6) that the OPT-125m model with LoRA, as compared with the pre-PEFT baseline, achieves much higher peak accuracy, positive and negative class precision, recall and F1-score, and as reinforced by the confusion matrix in Fig. 7., we now predict true negatives with greater accuracy. This may be due to random stratified sampling, or perhaps that the valence of negative tweets is more negative than that of the positive valence of positive tweets. It is possible that disingenuity or sarcasm is 'tricking' or negation, like "I ain't like no body," is confusing our model and contributing to inference error.

BitFit:

Applying BitFit PEFT to our baseline, we freeze the weights and cease to perform gradient update and instead simply update biases each iteration of training. As with LoRA, performance as illustrated by the learning curves for BitFitillustrated convergence at 3 epochs, so to conserve compute, all visualizations and tabulated results hereafter account for this.

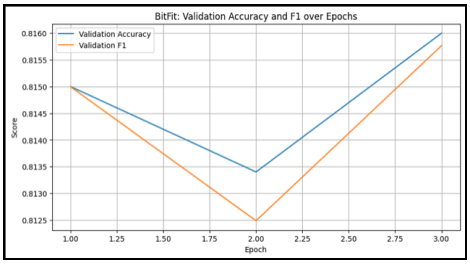

Fig 8: Learning curve of val acc and F1 on baseline OPT-125m w/ BitFit.

Fig. 8 appears to defy the model convergence claims on training, though accuracy and F1-score are bounded between 0.8125 and 0.8160, just 0.35% variation at just 3 epochs, hinting at convergence. The below figures illustrate (in comparison with the LoRA baseline) that BitFit achieves near the same validation accuracy (0.83 vs. LoRA's 0.84) while running training, on average, 60 seconds faster, likely due to it only updating biases, lending cheapness over LoRA.

OPT-125m w/ BitFit - Training Metrics							
	lr	batch_size	best_val_f1	val_accuracy	runtime_sec	memory_delta_mb	Test Accuracy
2	0.0001	8	0.831165	0.8312	1667.348322	0	0.826
0	0.00005	8	0.825385	0.8254	1675.313485	0.132812	
3	0.0001	16	0.815767	0.816	1607.933084	0	Test F1 (macro)
1	0.00005	16	0.804386	0.805	1606.963252	0.136719	0.826

Fig 9: Baseline OPT-125m w/ BitFit - summary training performance metrics.

OPT-125m Inference on Sentiment140 Dataset w/ BitFit (Preprocessed, Downsampled to 50k instances)				
Test Accuracy			0.8274	
F1 Score (macro):			0.827	
F1 Score (weighted):			0.827	
Inference Time:		[s]	28.51	
Classification Results				
	Precision	Recall	F1-Score	Support
Negative	0.86	0.78	0.82	2500
Positive	0.8	0.87	0.84	2500
Accuracy			0.83	5000
Macro Average	0.83	0.83	0.83	5000
Weighted average	0.83	0.83	0.83	5000

Fig 10: Baseline OPT-125m with BitFit - summary inference results.

Comparing Fig. 7 and 11, performance is indistinguishable. In this way, BitFit lends itself to the underlying objective of the experiment in discerning the cheapest and fastest means of model fine tuning on the binary sentiment analysis task.

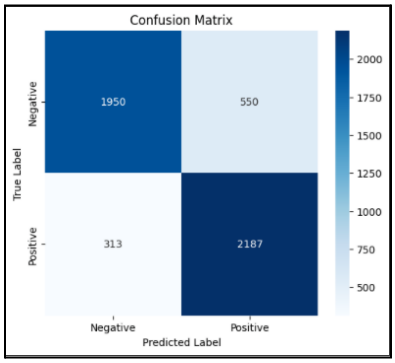

Fig 11: Confusion matrix of binary accuracy, baseline OPT-125m w/ BitFit.

Model Distillation Results:

For model distillation, we chose to work with Meta's larger OPT-1.5b. However, after experimentation, the model took significant training time (with and without LoRA) while having difficulty achieving nearly the same accuracy as our baseline results. With our current resources we did not have the compute necessary to perform further training and improve results.

We pivoted to Meta's OPT-350m parameter as the teacher. After training, the model was able to get about 1-1.5% better results than our baseline OPT-125m model. More importantly, we wanted to utilise its optimised internal representations for better outputs. As mentioned by Hinton et al. (2015), we hope that the probability distribution of the final classification of the teacher model carries more information that only a fully trained network would be able to convey [6]. This would "guide" the student to a better internal representation for itself, yielding a better more generalised approach that the model would not be able to develop on its own without significant additional training.

OPT-125m Inference on Sentiment140 Dataset (Distilled from OPT-350m, Full Dataset)				
Test Accuracy			0.8283	
F1 Score (macro):			0.8283	
F1 Score (weighted):			0.8281	
Inference Time:	[s]		80.9	
Classification Results				
	Precision	Recall	F1-Score	Support
Negative	0.82	0.84	0.83	12000
Positive	0.84	0.81	0.83	12000
Accuracy			0.83	24000
Macro Average	0.83	0.83	0.83	24000
Weighted average	0.83	0.83	0.83	24000

Fig 12: Inference results for OPT-350m distilled to OPT-125m (all data)

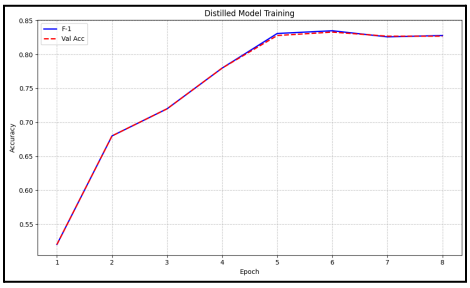

Fig 12: Distilled model learning curve -val acc and F1 Score. .

Since our goal for model distillation was not training efficiency but overall better results, we trained the teacher model on the full Sentiment140 dataset, with each epoch taking 1.5 hours to finish. We then run through epochs where the teacher model's outputs were being weighed into the training loss of the student model. The training dataset consisted, as before, of the full dataset.

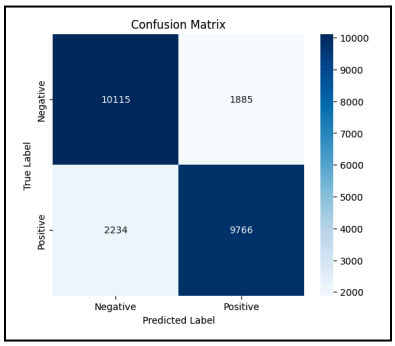

Fig 11: Confusion matrix for OPT-125m distillate (from OPT-350m).

CONCLUSION

The project explored the effectiveness of Low-Rank Adaptation (LoRA) and BitFit, two parameter-efficient fine-tuning (PEFT) techniques, on a binary sentiment classification task using the Sentiment140 dataset. After preprocessing and downsampling the dataset to 50,000 balanced and denoised tweets, we fine-tuned an OPT-125M model using both techniques and evaluated them on classification accuracy, training runtime, and memory usage. LoRA improved test accuracy from the baseline 0.53 to 0.84, while BitFit reached a slightly lower but still strong 0.83. While LoRA produced slightly better performance, noticeably faster, finishing training about 60 seconds faster per trial because of its lightweight bias-only updates and smaller memory footprint.

We also incorporated model distillation, first attempting to distill from OPT-1.5B, which proved infeasible due to extended training times and underwhelming performance. Shifting to OPT-350M as the teacher model, we achieved a 1-1.5% improvement in the student model's accuracy compared to baseline. The distillation process, though it is computationally intensive, demonstrated that student models can benefit from the representational knowledge of larger models. However, the training, which included both the student and teacher model, as well as the large size of the full dataset yielded relatively small gains. We hypothesize that for "simple" tasks such as binary classification, the outcome probabilities do not include much additional information from which the student model can learn. Larger, more complex problems such as multinomial sentiment classification would be more likely to yield substantial results. A larger teacher model would be able to infer many more subtle characteristics about the data that a smaller model would have trouble on its own.

Overall, our experiments confirm that both LoRA and BitFit offer viable paths for fine-tuning LLMs under limited-resource constraints, with LoRA providing slightly better accuracy and BitFit excelling in efficiency. The combination of PEFT with model distillation presents a promising direction for building high-performing yet lightweight NLP models.

REFERENCES

[1] A. Sharma, "Efficient Fine-Tuning for Large Language Models: Maximizing Results with Limited Resources," *Medium*, Jan. 18, 2024. [Online]. Available: https://medium.com/@akshitsharma105/efficient-fine-tuning-for-large-language-models-maximizing-results-with-limited-resources-72e7acf3a591 [Accessed: Mar. 3, 2025].

[2] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Report, vol. 1, no. 2009, p. 12, 2009. [Online]. Available: https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf. [Accessed: Mar. 3, 2025].

[3] R. Rust *et al.*, "How Effective is Parameter-Efficient Fine-Tuning? A Systematic Benchmark," *arXiv preprint*, 2023. [Online]. [Accessed: Mar. 2, 2025].

[4] Lester, B., Al-Rfou, R., and Constant, N. "The Power of Scale for Parameter-Efficient Prompt Tuning." *arXiv preprint*, arXiv:2104.08691, 2021. [Online]. Available: https://arxiv.org/abs/2104.08691. [Accessed: Mar. 3, 2025].

[5] S. Zhang, S. Roller, N. Goyal, et al., "OPT: Open Pre-trained Transformer Language Models," *arXiv preprint arXiv:2205.01068*, 2022. [Online]. Available: https://arxiv.org/abs/2205.01068

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531. [Accessed: Mar. 28, 2025].

[7] J. Henderson, P. Tamilselvan, S. Allaudin, A. Taprantzis., "Sentiment Analysis on Parameter-Efficient Neural Distillation with Iterative Fine-Tuning," Project Proposal, *CS 7650: Natural Language Processing*, Georgia Institute of Technology, Atlanta, GA, USA, Mar. 2025.

[8] P. Tamilselvan, "SEND-IT," GitHub. [Online]. Available: https://github.com/Priya-753/SEND-IT. [Accessed: Apr. 25, 2025].