Ng Rui Bin, Damien
(A0252828X)

DSA4213 Assignment 3 Report

Context

I chose a base model DistilBERT-base-uncased, to do a domain specific (financial sentiment analysis) classification task. My first model is the full fine-tuning DistilBERT-base-uncased, while my next parameter-efficient tuning model will be with LoRA fine-tuning.

I will be evaluating my classification model using accuracy, F1 (Macro), F1 (Weighted), and the training time.

Dataset Selection and Motivation

The dataset that I selected was "zeroshot/twitter-financial-news-sentiment" from HuggingFace (https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment). This dataset consists of an annotated corpus of finance-related tweets, which can be used to classify finance-related tweets for their sentiment.

The motivation for this dataset is:

- Practical and Well-researched NLP task: financial sentiment analysis is practical and well-researched, and has clear real-world applications such as market trend predictions and trading strategies
- Size, diversity, and availability: the dataset is balanced, labelled, and publicly available, making it ideal for reproducible fine-tuning experiments

Model and Fine-tuning Strategies Used

The model I used for my full fine-tuning is the DistilBERT-base-uncased model. I used this model because DistilBERT offers a good trade-off between speed, resource efficiency, and accuracy, where even though its 40% smaller than BERT, its 60% faster, and retains ~97% of performance.

For my full fine-tuning, all the model weights were updated during training, with 100% trainable parameters.

For my LoRA fine-tuning, only low-rank adapter layers were trained efficiently, with only 1.09% trainable parameters. Also, for the LoRA configuration:

- Rank = 8
- Alpha = 16
- Dropout = 0.1
- Applied to q_lin and v_lin attention layers only

The data split was split into train, validation, test, 70%, 15%, 15% respectively. The table below is the training hyperparameters for both full fine-tuning and LoRA model.

Parameter	Full Fine-Tuning	LoRA

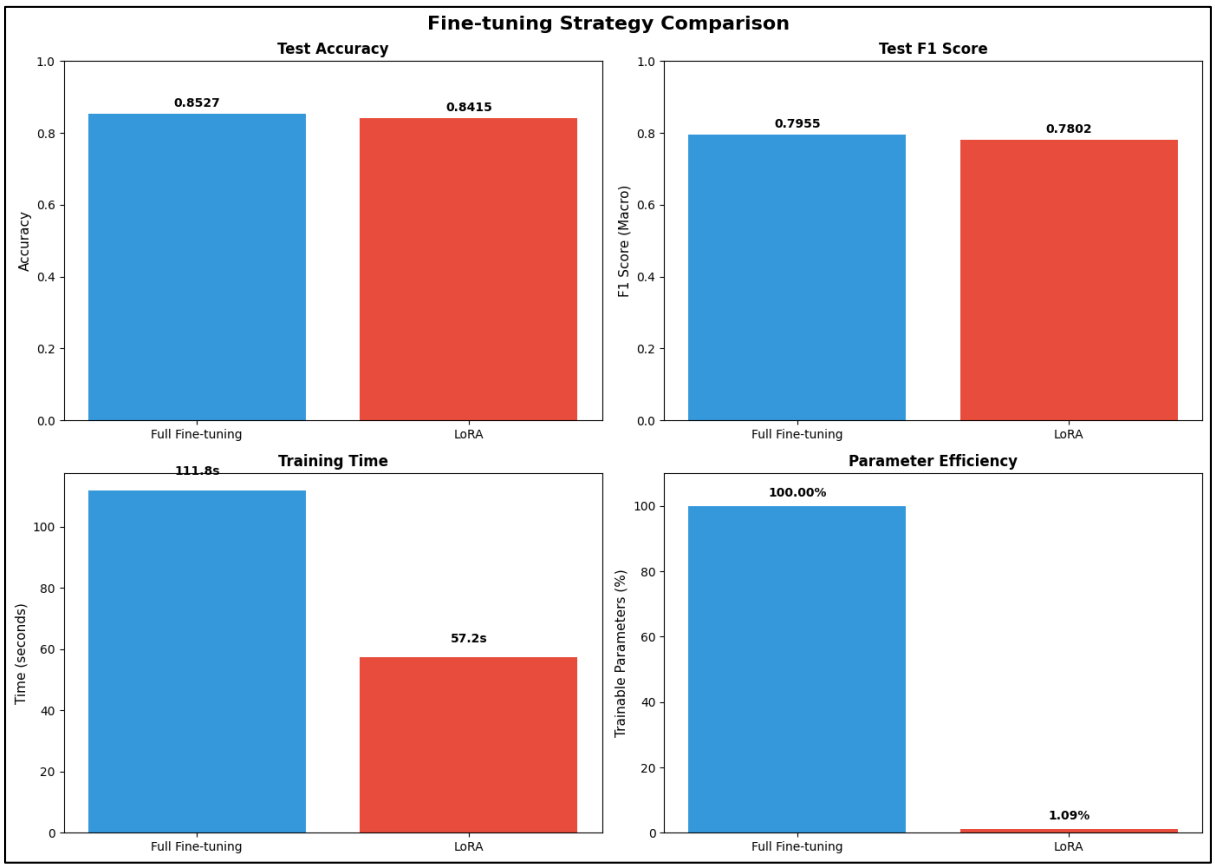
Ng Rui Bin, Damien
(A0252828X)

Learning Rate	2e-5	3e-4
Batch Size	16	16
Epochs	3	3
Weight decay	0.01	0.01
Warmup steps	100	100

For evaluation, I used accuracy, F1 (Macro), F1 (Weighted), and train time as the evaluation metrics. The results based on evaluation on the test set can be seen in the table below.

Model	Accuracy	F1 (Macro)	F1 (Weighted)	Training Time (s)	Trainable Parameters
Full Fine-Tuning	0.8595	0.7955	0.8525	111.77	66,955,779
LoRA	0.8415	0.7802	0.8411	57.21	740,355

The results are summarised in the plot below.



Ng Rui Bin, Damien
(A0252828X)

Key Takeaways

Firstly, full fine-tuning provides higher accuracy (85.95%) compared to LoRA (84.15%). However, this requires updating all the parameters.

Next, LoRA, which is a form of parameter-efficient tuning, actually achieves competitive results while training ~1% of the total parameters, which means that there is a parameter efficiency gain.

Next, LoRA models are faster, lighter, and more deployable on limited hardware, especially with no GPU, compared to training a full fine-tuning model.

Limitations

Firstly, the dataset is considered short and noisy texts. As they are tweets, it is brief, and contain informal language (e.g. "$TSLA"). These features would introduce noise, and make it difficult for the model to extract precise sentiment without context.

Secondly, financial tweets are isolated extracts that does not include the full story and context. The tweet might mention something like "earnings call" or "market reaction", referencing an event that happened earlier, or a linked article that is not included in the dataset. This lack of continuity will limit the model's understanding of why sentiment is positive or negative.

Thirdly, sentiment labels in financial contexts can be ambiguous. A tweet like "Stock dips before earnings, good entry point?" could be labelled as negative or positive, depending on the investor's perspective. This would introduce label noise and reduce the accuracy.

Lastly, sudden events such as USA vs China trade tariffs can temporarily shift sentiment distributions. Static models cannot dynamically adapt to these high-volatility linguistic patters unless retrained periodically.

Beyond the minimum requirements

Error analysis

Some of the tweets are filled with sarcasm (e.g. "Great, another record low for $TSLA"). These tweets were often misclassified as positive due to the presence of "positive" words such as "great".

There might be mixed sentiments regarding tweets that talks about both sides of the situation (e.g. "bullish short term, but wary of Fed rates"). This would confuse the model as the first half is positive, but the second half is negative.

Future Work

Here, I will mention some of the possible future work that can be done.

Ng Rui Bin, Damien
(A0252828X)

Firstly, we can have hybrid modelling. We could combine textual sentiment models with numerical market data (e.g. price change, volume change) to better ground sentiment in measurable financial outcomes. This is because financial sentiment does not exist alone, where tweets often reference stock prices, indices or macroeconomic indicators. We can use a text encoder alongside tabular features in a simple MLP or transformer-based architecture, in hopes of improving accuracy in tweets with ambiguous sentiment, and help the model distinguish between perceived sentiment and actual market response.

Next, we can do dynamic learning. As financial language evolves rapidly, and new financial instruments appear in the market frequently, static models would quickly become outdated. We should implement continuous learning pipelines where the model can periodically learn new labelled or pseudo-labelled financial tweets. To prevent the model from forgetting, we can use techniques like LoRA or adapter modules that would allow updating only a small set of parameters. This is so as to keep the model aligned with the current trends and instruments, improving robustness of the model.