LLM-Based Financial Sentiment Analysis for Trading Signals

EN.553.640(01&02) Machine Learning in Finance Final Project Final Report

[Guojun Peng, Fusheng Luo, Jiamu Chen, Yining Li]

1 Introduction

1.1 Problem Background

Large Language Models (LLMs) are increasingly prominent in financial text analysis, allowing the extraction of predictive signals from unstructured sources such as earnings calls, news articles, and regulatory filings. Their deep understanding of context and sentiment surpasses traditional dictionary-based methods. However, applying general-purpose LLMs to financial sentiment analysis presents two key limitations: (1) they are not trained on finance-specific language and may misinterpret domain-specific terms, and (2) they require immense computational resources for fine-tuning, restricting accessibility.

Recent work has adapted LLMs for finance by fine-tuning models like LLaMA on financial news or disclosures to improve return prediction. Some studies rely on full-model optimization, which demands substantial computational resources. Others adopt parameter-efficient approaches, such as LoRA, updating only a small subset of weights to reduce memory cost. Although both strategies have shown strong empirical performance, they still leave room for further balance of efficiency and predictive power, especially for practitioners with limited access to high-end hardware.

This project explores the application of Quantized LoRA (QLoRA), a recent fine-tuning method that combines 4-bit quantization with low-rank adaptation to significantly reduce memory usage without sacrificing performance. By freezing the pre-trained model weights in 4-bit precision and backpropagating through 16-bit LoRA adapters, QLoRA enables the efficient fine-tuning of models with up to 65 billion parameters on a single 48GB GPU (Dettmers et al., 2023).

Although QLoRA has shown impressive results in general NLP tasks, its effectiveness in financial applications remains largely untested. In this project, our objective is to evaluate whether QLoRA can maintain competitive predictive performance in generating alpha signals from financial news sentiment. Using finance-specific LLaMA-3 models and labeled data sets such as Financial PhraseBank or NASDQ news sentiment, we will assess signal quality through backtesting metrics, including Sharpe ratio and cumulative returns.

1.2 Related Literature

Beyond finance, a large-scale benchmark by Justin Zhao [1] evaluates LoRA-based fine-tuning across 310 large language models (LLMs), covering 31 tasks and 10 different base models. Their findings show that LoRA-tuned 4-bit models consistently outperform their base versions, and even GPT-4 on certain tasks, while requiring significantly less memory. Additionally, they introduce LoRAX, an inference server capable of serving dozens of LoRA adapters simultaneously on a single GPU by sharing base model weights. This demonstrates the effectiveness and efficiency of LoRA-based fine-tuning for various downstream applications.

Within the financial domain, Giorgos Iacovides [2] proposed FinLLama, a framework that fine-tunes LLaMA-2 in labeled financial sentiment data to generate sentiment scores and trading signals. Their approach employs LoRA and 8-bit quantization to reduce computational cost while maintaining strong portfolio performance [3]. However, their method still relies on moderately sized models and does not explore deeper quantization or more lightweight deployment strategies.
To address the resource constraints of large-scale LLM fine-tuning, Tim Dettmers [4] recently introduced QLoRA, a highly memory-efficient extension of LoRA. QLoRA combines 4-bit NormalFloat quantization with LoRA adapters and techniques such as paged optimizers and double quantization to enable full 16-bit training quality on a single 48GB GPU. Their models, including the Guanaco family, achieve performance near that of ChatGPT on the Vicuna benchmark, offering a scalable and accessible alternative for fine-tuning large models in low-resource settings.

2 Problem Setup

We aim to develop a sentiment classification model tailored to financial news articles using a fine-tuned LLaMA model. The task is formulated as a supervised learning problem where each input is a piece of financial text, and the output is a discrete sentiment label indicating the article's sentiment regarding financial markets or assets.

Problem Definition: Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the training dataset, where each $x_i \in \mathcal{X}$ represents a financial news excerpt and $y_i \in \mathcal{Y}$ is the associated sentiment label. The label set is defined as

$$\mathcal{Y} = \{-1, 0, 1\}$$

where:

- -1 denotes negative sentiment,
- 0 denotes neutral sentiment,
- 1 denotes positive sentiment.

The goal is to learn a function $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ (the model parameters) that minimizes the expected classification error. Formally, we seek:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}(f_\theta(x), y)\right]$$

where $\mathcal{L}$ is a loss function, here we adopt the cross-entropy loss for multi-class classification.

Model Setup: We employ a pre-trained LLaMA model as the backbone, fine-tuned with LoRA (Low-Rank Adaptation) techniques to enable efficient parameter tuning. During fine-tuning, we adapt a small subset of parameters while freezing the majority of the pre-trained weights, thereby retaining general language understanding while specializing the model for financial sentiment detection.

Assumptions:

- The labeled dataset $\mathcal{D}$ is representative of the financial news distribution we expect the model to encounter during inference.
- Sentiment expressed in financial news can be reasonably categorized into the three discrete classes: negative, neutral, and positive.
- The fine-tuning process does not significantly degrade the base LLaMA model's language understanding capabilities outside the financial domain.

3 Dataset

3.1 Raw Data

We have divided the process of our model building into two parts: fine-tuning the pre-trained Large Language Models (LLM) and using the post-trained model to predict the sentiment of our News dataset and conducting trading strategies combined with the dataset of stocks price in the US equity markets.

3.1.1 Part1: Fine-tuning Dataset

We generally use four datasets here

[label=()]Financial PhraseBank-v1.0 NASDQ news sentiment Twitter Financial News Sentiment FIQA2018

(i)

Financial Phrase Bank dataset covers a collection of 4840 sentences. The selected collection of phrases was annotated by 16 people with adequate background knowledge on financial markets. Three of the annotators were researchers and the remaining 13 annotators were master's students at the School of Business at Aalto University with majors primarily in finance, accounting, and economics. They have labeled the News with sentiments of **positive (+1) , negative (-1) and neutral (0)**. E.g., The news "The operating margin came down to 2.4 from 5.7." is labeled as **negative**

(ii)

The "NASDQ news sentiment dataset" consists of synthetic data points generated using GPT-4o, and can be used to train models for analysing sentiment from the title and meta-description of an article. This can be used for tasks such as analyzing market sentiment regarding a particular NASDAQ-listed stock from search results. GPT has also labeled the news articles with positive, negative and neutral sentiments

(iii)

The Twitter Financial News dataset is an English-language dataset containing an annotated corpus of finance-related tweets. This data set is used to classify financial-related tweets according to their sentiment.The data set contains 11,932 documents annotated with 3 labels: **label 0: Bearish; label 1: Bullish, label 2: Neutral**

(iv)

FIQA2018 dataset (often stylized as FiQA or FiQA: A Challenge on Financial Opinion Mining and Question Answering) refers to a dataset and shared task introduced for the WSDM Cup 2018. It is designed to benchmark financial domain sentiment analysis. It also classifies news sentiments into positive and negative. But it differs in that they give a **sentiment score from -1 to +1** to measure the sentiment of the News ("+1" refers to highly positive, and "-1" refers to highly negative). A score of 0 indicates neutral or without clear polarity.

3.1.2 Part2: Predicting Real-Time News

In this section, we use only two datasets, **the news (which includes news articles and alerts, the alerts denote the news with only headlines) dataset and the corresponding stock price dataset**. We collected news text data from Refinitiv (now known as LESG) and CRSP &

YahooFinance for stock prices. Then we create a news dataset covering the lists of stocks within the time range of 13 months (2024.3.11 - 2025.4.23). This news dataset contains 305 stocks and 69494 news headlines.

3.2 Data Preprocessing

The preprocessing steps applied to the raw news data obtained from Refinitiv (LSEG) are as follows. Due to the data extraction constraint of 100 records per request, we adopt a recursive collection approach by dividing the full time horizon into consecutive two-week intervals. This strategy allows us to systematically retrieve news for each stock of interest over the entire sample period. After compiling the data across all intervals, we concatenate them into a unified dataset, append the relevant stock ticker symbols, and set the date as the index to match the structure required for our daily long-short trading strategy. To ensure consistency and prevent overcounting, we also remove any duplicated records.

3.2.1 Label Mapping

To enable consistent sentiment classification across diverse datasets, we apply a standardized label mapping function that transforms the original dataset-specific sentiment annotations into a unified label space of {-1, 0, 1}, corresponding to *negative*, *neutral*, and *positive* sentiments, respectively. The mapping rules for each dataset are as follows:

- **Financial PhraseBank**: The original labels are textual strings: `"negative"`, `"neutral"`, and `"positive"`. These are mapped directly via:

  $$\texttt{"negative"} \to -1, \quad \texttt{"neutral"} \to 0, \quad \texttt{"positive"} \to 1$$

  Any unexpected label is defaulted to 0 (neutral).

- **FIQA 2018**: The dataset provides a continuous sentiment score in the range $[-1.0, 1.0]$. We discretize it as follows:

  $$\text{score} < -0.2 \to -1, \quad \text{score} > 0.2 \to 1, \quad \text{otherwise} \to 0$$

- **Twitter Financial Sentiment**: Labels are originally encoded as integers: `0` (negative), `1` (positive), and `2` (neutral). These are remapped to:

  $$0 \to -1, \quad 2 \to 0, \quad 1 \to 1$$

- **NASDAQ News**: The sentiment is expressed as an integer score, typically ranging from 0 to 5. The mapping is defined as:

  $$\text{score} \leq 1 \to -1, \quad \text{score} \geq 4 \to 1, \quad \text{otherwise} \to 0$$

This label normalization procedure ensures all training samples conform to the same target format, enabling consistent training and evaluation across datasets.

3.2.2 Tokenizer

The tokenizer uses the official vocabulary and tokenization rules for LLaMA 3. It maps raw input to token IDs aligned with the model's embedding space.

4 Methodology

4.1 QLoRA

4.1.1 LoRA Model

Low-rank adaptation (LoRA) [5] is a parameter-efficient finetuning method that preserves the pretrained transformer model weights and introduces a smaller set of trainable weights, which are expressed using low-rank decomposition.
In LoRA, the update weights are assumed to follow the low-rank decomposition:

$$\Delta W = BA, \tag{1}$$

where $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{n\times r}$ are trainable parameters, and $W_0 \in \mathbb{R}^{n\times n}$ is the pretrained weight matrix. Note that n can be large (e.g., $n = 4096$) and the rank $r \ll n$, typically $r = 4, 8$, or 16. For example, setting $n = 4096$ and $r = 8$, W_0 has about 16 million parameters, while A and B together have 65,536 parameters, which is about 0.039% of the size of W_0.
During the finetuning stage, the forward pass is:

$$y = W_0x + \Delta Wx = W_0x + BAx, \tag{2}$$

where W_0 denotes the pretrained weights.

During the inference stage, A and B are merged back into W_0, resulting in the matrix W:

$$W = W_0 + BA, \tag{3}$$

$$y = Wx. \tag{4}$$

Therefore, LoRA does not introduce additional costs to the inference process.

4.1.2 Quantile

Using the components described above, we define QLoRA [6] for a single linear layer in the quantized base model with a single LoRA adapter as follows:

$$\mathbf{Y}^{\text{BF16}} = \mathbf{X}^{\text{BF16}} \cdot \text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{NF4}}) + \mathbf{X}^{\text{BF16}}\mathbf{L}_1^{\text{BF16}}\mathbf{L}_2^{\text{BF16}}. \tag{5}$$

The `doubleDequant` function is defined as:

$$\text{doubleDequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}, \mathbf{W}^{\text{k-bit}}) = \text{dequant}(\text{dequant}(c_1^{\text{FP32}}, c_2^{\text{k-bit}}), \mathbf{W}^{\text{4bit}}) = \mathbf{W}^{\text{BF16}}. \tag{6}$$

We use NF4 for $\mathbf{W}$ and FP8 for c_2. A blocksize of 64 is used for $\mathbf{W}$ for higher quantization precision, and a blocksize of 256 is used for c_2 to conserve memory.
For parameter updates, only the gradient with respect to the adapter weights $\frac{\partial E}{\partial \mathbf{L}_i}$ is required—not the gradient with respect to the 4-bit weights $\frac{\partial E}{\partial \mathbf{W}}$. However, computing $\frac{\partial E}{\partial \mathbf{L}_i}$ still requires calculating $\frac{\partial \mathbf{X}}{\partial \mathbf{W}}$, which is done via Equation (5) using dequantized $\mathbf{W}^{\text{BF16}}$.

To summarize, QLoRA uses one storage data type (usually 4-bit NormalFloat) and one computation data type (16-bit BrainFloat). The storage data type is dequantized into the computation data type for both the forward and backward pass. However, weight gradients are computed only for the LoRA parameters using 16-bit BrainFloat.

4.2 Swift

SWIFT is an open-source [7], unified framework designed to support the full lifecycle of Large Language Models (LLMs) and Multi-modal Large Language Models (MLLMs), including training, fine-tuning, evaluation, and deployment. It offers built-in support for over 300 LLMs and 50 MLLMs, and integrates a wide range of parameter-efficient tuning methods such as QLoRA, LoRA+, and LongLoRA. SWIFT supports both supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF), and includes utilities for dataset preprocessing, quantization, inference, and benchmark evaluation.

A key feature of SWIFT is its modular and extensible architecture, which unifies pure-text and multi-modal model development through a standardized interface for models, datasets, training configurations, and deployment targets. It enables users to train large-scale models efficiently on limited hardware through techniques like low-bit quantization and gradient checkpointing. With command-line and web-based interfaces, SWIFT simplifies the end-to-end adaptation of foundation models for downstream tasks across text and vision domains.

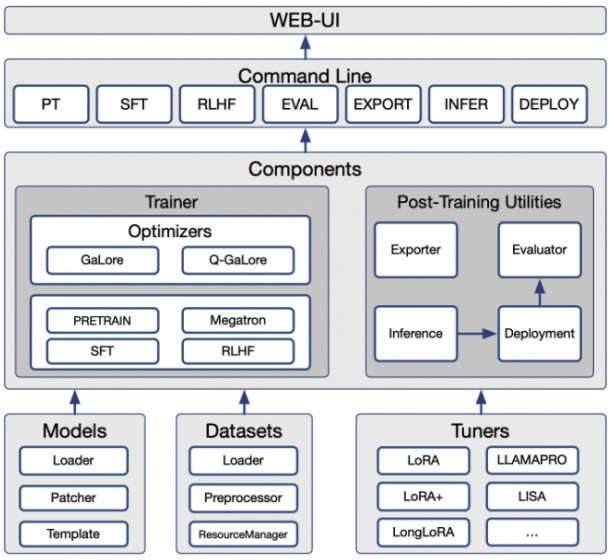

Figure 1: The frame work of SWIFT.

4.2.1 Swift Trainer

We adopt the SWIFT training framework in our project due to its streamlined integration of large language model fine-tuning techniques. While the Hugging Face ecosystem—including Transformers, PEFT, Accelerate, and BitsAndBytes—offers extensive support for fine-tuning, parameter-efficient adaptation (e.g., LoRA, QLoRA), and low-bit quantization, implementing these components together often requires separate configurations and dependencies. In contrast, SWIFT pro-

vides a unified interface that consolidates these features into a single, user-friendly training pipeline.

Specifically, SWIFT simplifies the application of 4-bit quantization, gradient checkpointing, and FlashAttention-style optimizations, which are crucial for training large models with limited resources. Furthermore, SWIFT facilitates end-to-end training by integrating evaluation, deployment, and export functionalities into its core interface. These capabilities reduce the engineering overhead required to prepare and execute efficient LLM fine-tuning, allowing us to focus on model quality and domain adaptation.

4.2.2 Parameters

Low-Rank Adaptation (LoRA) Configuration: LoRA parameters are set with a rank $r = 16$, `lora_alpha` $= 32$, and `lora_dropout` $= 0.1$.

Learning Rate and Optimizer: The model is fine-tuned using a learning rate of 1×10^{-4} with the AdamW optimizer.

Batch Size and Epochs: Training is conducted over 1 epoch with a batch size of 8 and gradient accumulation set to 4.

Training Details: We set the ratio 9:1 of the train set to the test set; Gradient checkpointing is enabled to reduce memory usage during backpropagation, which facilitates training with limited GPU resources. A cosine learning rate scheduler is used to gradually decay the learning rate over time, enhancing convergence and stability.

4.3 Trading Strategy

To evaluate the practical profitability of our sentiment classification model, we design a straightforward portfolio strategy based on daily news sentiment predictions [9]. Specifically, for each stock, we aggregate all predicted sentiment labels from the daily news and compute the average sentiment score. Using this score, we construct a zero-net-investment portfolio by taking equal-weighted long positions in the top quintile (15%) of stocks with the most positive sentiment and short positions in the bottom quintile (15%) with the most negative sentiment. The portfolio is rebalanced on a daily basis to reflect updated sentiment signals.

5 Experiments and Results

5.1 Experiments

We use Google Colab to train and test our model. We train our model based on four financial text datasets, which cost about 18 hours, with 90 computer units. Then we compare the classification precision with another financial sentiment llm, FinBert. The results shows that our model can classify the sentiment more precisely. The portfolio return also reflects that our model is profitable.

5.2 Results

5.2.1 Classification Performance

F1 Score Overview

The F1 score is a widely used metric for evaluating classification performance, especially in imbalanced datasets. It is defined as the harmonic mean of precision and recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Precision measures how many of the predicted positive instances are actually correct. It is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{8}$$

A high precision indicates a low false positive rate.

Recall (also called sensitivity or true positive rate) measures how many of the actual positive instances are correctly identified by the model. It is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{9}$$

A high recall indicates that most relevant instances are captured.

There are two common approaches to aggregate F1 scores across multiple classes:

- **Macro F1 Score:** Computes the F1 score independently for each class and then takes the unweighted average. It treats all classes equally, regardless of their support (number of true instances).
- **Micro F1 Score:** Aggregates the contributions of all classes to compute the global precision and recall, and then derives the F1 score. It gives more weight to the performance on common classes.

We evaluated our fine-tuned Meta-LLaMA3-8B sentiment classifier on a test set of 9,064 financial news samples. Our model achieved an impressive overall accuracy of **92.18%**, with a macro-average F1 score of **0.5787** and a micro-average F1 score of **0.9218**. The class-wise performance reveals that the model performs exceptionally well on `Positive` and `Neutral` samples, with F1 scores of **0.9636** and **0.7172**, respectively. However, its performance on the `Negative` class remains modest (F1 = 0.0552), likely due to class imbalance within the training data.

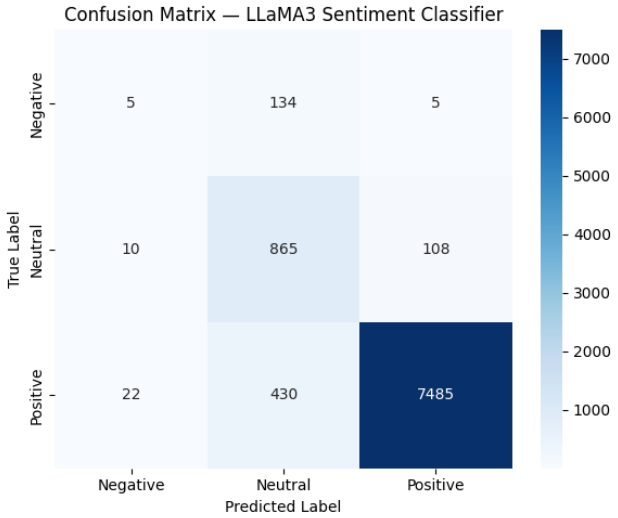

Figure 2: Confusion Matrix of Tuned-LLaMA3 Sentiment Classifier.

Tuned-LLaMA3 Classifier

Class	Precision	Recall	F1-score	Support
Negative	0.1351	0.0347	0.0552	144
Neutral	0.6053	0.8800	0.7172	983
Positive	0.9851	0.9431	0.9636	7937

Overall Results (Tuned-LLaMA3):

- Accuracy: **92.18%**
- F1 Score (macro): **0.5787**
- F1 Score (micro): **0.9218**

For baseline comparison, we evaluated FinBERT [11] on the same test set. While FinBERT attained high precision for the `Positive` class (0.8232), its overall recall was significantly lower, resulting in an F1 score of only **0.5316** for that class. Performance on the `Negative` and `Neutral` classes was notably poor, with an overall accuracy of just **36.26%** and macro F1 of **0.2101**.

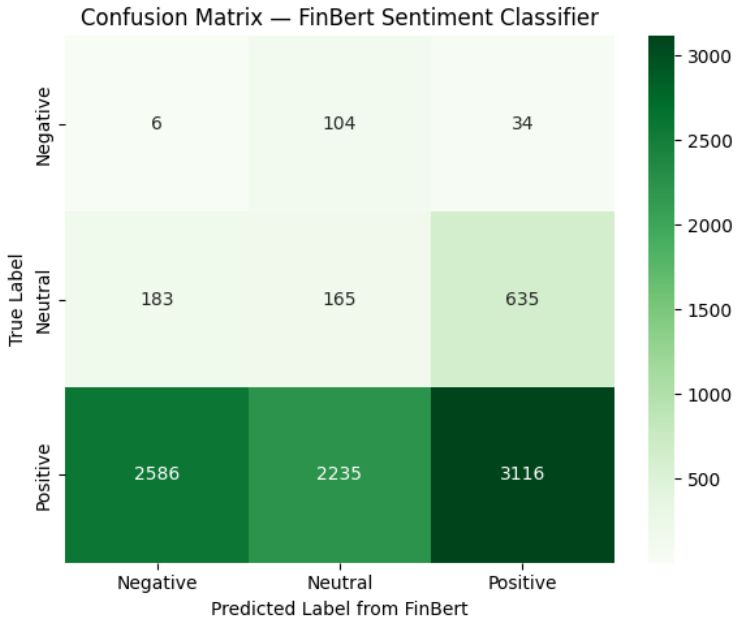

Figure 3: Confusion Matrix of FinBERT Sentiment Classifier

FinBERT (Baseline)

Class	Precision	Recall	F1-score	Support
Negative	0.0022	0.0417	0.0041	144
Neutral	0.0659	0.1679	0.0946	983
Positive	0.8232	0.3926	0.5316	7937

Overall Results (FinBERT):

- Accuracy: **36.26%**

- F1 Score (macro): **0.2101**
- F1 Score (micro): **0.3626**

The superior performance of our LLaMA3-based classifier over FinBERT demonstrates the effectiveness of domain-specific fine-tuning on diverse financial datasets. Our model not only generalizes better across sentiment classes, especially for neutral sentiment, but also avoids the strong positive bias shown by FinBERT. Although the Negative class remains underrepresented and underperformed, the overall results indicate that our fine-tuned LLaMA3 model significantly surpasses FinBERT in both accuracy and robustness, making it a strong candidate for downstream financial sentiment analysis tasks.

5.2.2 Portfolio Performance

Trading Strategy

To assess the real-world applicability of our sentiment classifier [10], we implement a daily rebalanced long-short trading strategy. On each trading day, sentiment signals are aggregated across stocks. If there are at least 20 valid signals available, stocks are ranked by their average sentiment score.

We take long positions in the top 15% of stocks with the most positive sentiment and short positions in the bottom 15% with the most negative sentiment. Each position is equally weighted, and the portfolio is fully rebalanced at the close of each trading day. The pseudo-code is followed.

Algorithm 1 Daily Sentiment-Based Portfolio Construction

for each day t **do**
  Extract all sentiment signals: `sent_pivot.loc[date]`
  if number of valid (non-NaN) scores ≥ 20 **then**
    Sort stocks by sentiment score
    Select:
    - Top 15% $\rightarrow$ long \$1
    - Bottom 15% $\rightarrow$ short \$1

    Update `current_longs` and `current_shorts`
  end if
  Calculate P&L on day $t + 1$
end for

Individual Performance

Tuned-LLaMA3 (Ours)

Our fine-tuned,Meta-LLaMA3-8B model on multiple financial sentiment datasets using QLoRA, produced the strongest trading performance, achieving an annualized return of **9.85%** and a Sharpe ratio of **1.03**. This demonstrates its ability to generate sentiment signals that align well with future price movements. The model benefits from being trained on multiple financial datasets, improving its generalization and reducing overfitting. It shows consistency and robustness, making it suitable for practical deployment in financial sentiment-based trading.

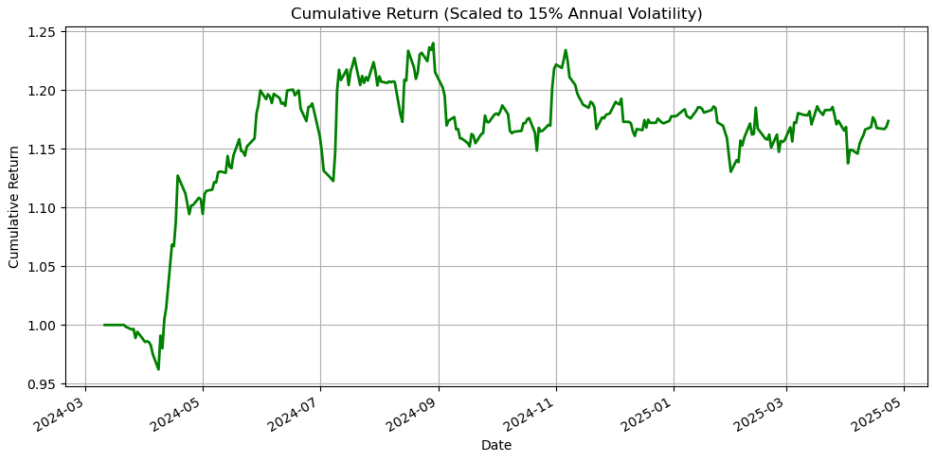

Figure 4: Cumulative Return — Tuned-LLaMA3

ProsusAI/FinBERT

ProsusAI's FinBERT model [8], a financial sentiment model, achieved a modest annualized return of **0.77%** and a Sharpe ratio of **0.13**. While its sentiment signals are somewhat predictive, the performance is not economically significant. This may be due to model rigidity or a lack of adaptation to recent market narratives. It performs better than other FinBERT variants, but still lags behind our fine-tuned model.
Link for finbert

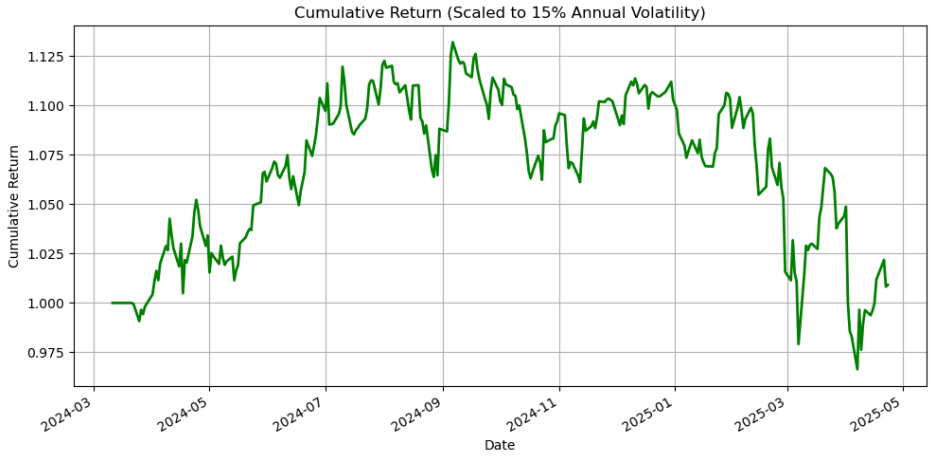

Figure 5: Cumulative Return — ProsusAI/FinBERT

FinancialBERT (ahmedrachid)

This is a BERT model trained on financial texts with sentiment analysis capabilitie. It's predictin scores produced an annualized return of **-10.09%** and a Sharpe ratio of **-0.99**. These negative values indicate that its sentiment signals are not only uninformative but potentially harmful if used directly for trading. The results may stem from misalignment between training data and current market tone.
Link for FinancialBERT

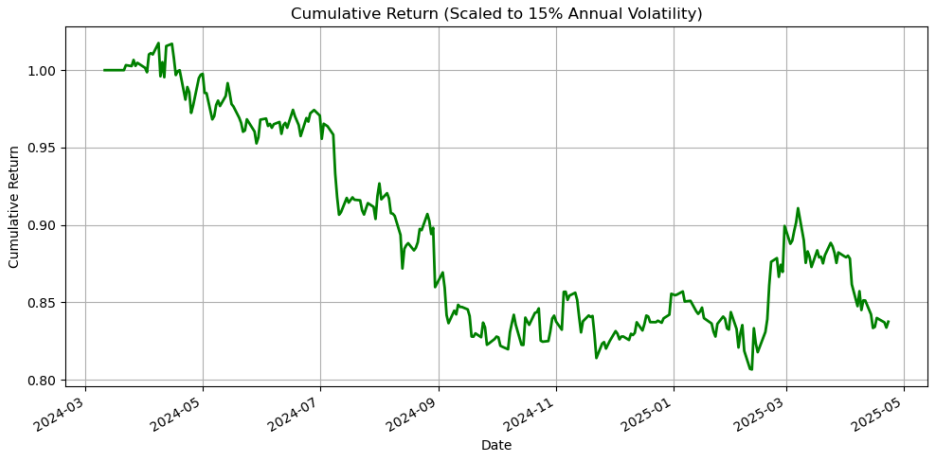

Figure 6: Cumulative Return — FinancialBERT (ahmedrachid)

FinBERT-Tone (yiyanghkust)

This model is a FinBERT variant focusing on tone-based sentiment signals. Its portfolio produced the weakest results, with an annualized return of **-16.89%** and a Sharpe ratio of **-2.02**. The sentiment outputs seem to exhibit systematic noise or directional error when applied in a trading setting. These results suggest a poor signal-to-noise ratio and highlight the importance of dataset relevance and fine-tuning.
Link for FinBERT_tone

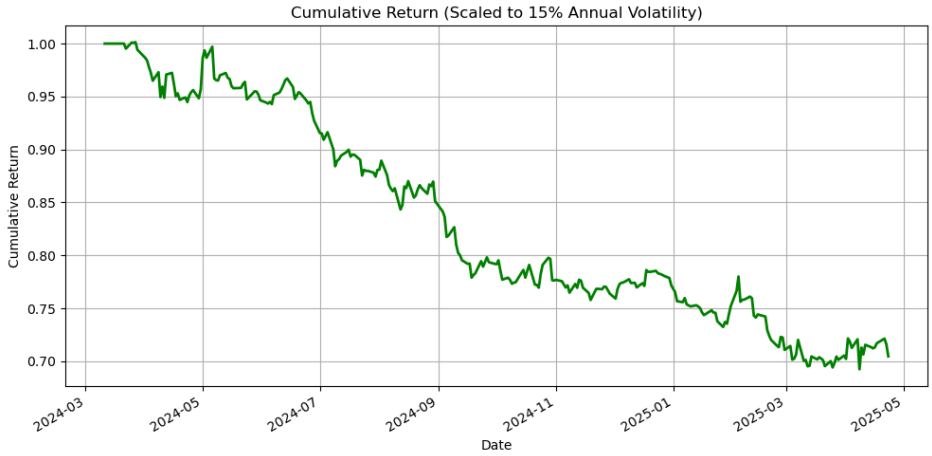

Figure 7: Cumulative Return — FinBERT-Tone (yiyanghkust)

Performance Comparison

We evaluate each model's signal based on the portfolio's annualized return and Sharpe ratio under the described trading strategy. And we make the results in one chart to directly show our model's profitable prediction result.

Table 1: Performance Comparison of Sentiment-Driven Portfolios

Model	Annualized Return	Sharpe Ratio
Ours (Tuned-LLaMA3)	9.85%	1.03
ProsusAI/finbert	0.77%	0.13
FinancialBERT (ahmedrachid)	-10.09%	-0.99
FinBERT-tone (yiyanghkust)	-16.89%	-2.02

The following plot compares the cumulative returns of the four strategies over time. Clearly, our fine-tuned LLaMA3 model outperforms all FinBERT variants baselines by a wide margin in both return and risk-adjusted terms, with substantially higher return and stability, confirming the effectiveness of domain-specific adaptation and recent data alignment.

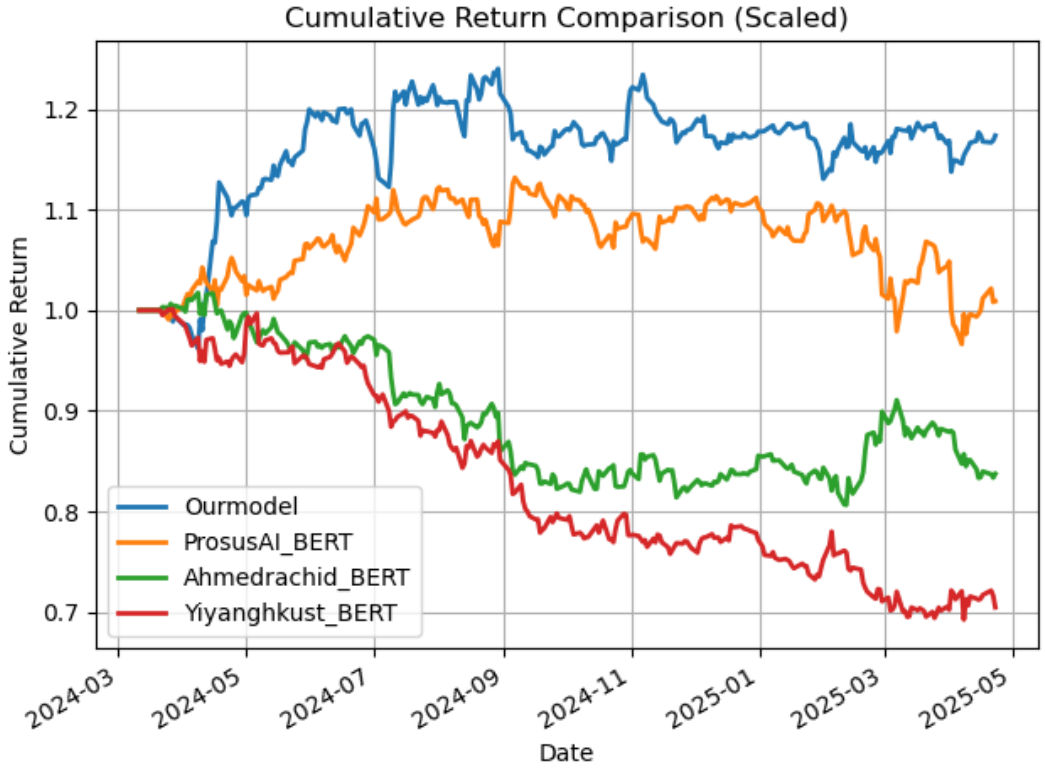

Figure 8: Cumulative Portfolio Returns: Our Model vs. FinBERT Variants

6 Conclusion

In this project, we explored the application of parameter-efficient fine-tuning techniques for financial sentiment analysis, specifically focusing on Quantized LoRA (QLoRA) applied to the Meta-LLaMA3-8B model. By integrating multiple financial sentiment datasets and employing the SWIFT framework, we successfully trained a domain-adapted model capable of generating high-quality sentiment signals from financial news headlines.

Our classifier demonstrated superior classification performance compared to FinBERT baselines, achieving a micro F1 score of 0.9218 and a macro F1 score of 0.5787, much better than baseline

model's. When applied to a long-short equity trading strategy, our model produced an annualized return of 9.85% and a Sharpe ratio of 1.03—substantially outperforming all FinBERT variants, some of which yielded negative returns and Sharpe ratios.

These results highlight the effectiveness of QLoRA-based fine-tuning in low-resource environments and validate the importance of domain-specific training for sentiment-driven financial forecasting. Our work suggests that using QLoRA-tuned LLaMA models can be powerful tools for extracting actionable insights from unstructured financial text, providing both statistical and economic value.

Future research may extend this framework to generate continuous sentiment scores rather than discrete classification labels, enabling finer granularity in signal generation. Additionally, incorporating broader market context features and leveraging reinforcement learning to directly optimize sentiment-driven trading strategies represent promising directions for enhancing both predictive accuracy and economic utility.

7 Elaboration on members' contributions to the project

All group members have invested significant effort into this project. Specifically, Fusheng LUO is responsible for the fine-tuning of models, training, and performance analysis. He also plays a crucial role in developing the trading algorithms for signals generated by our four different models, including our model versus BERT variants. Guojun Peng leads the whole research. He is in charge of literature synthesis and Research Question Development, training part of LLM models and building and backtesting trading strategy. LI also contributed to text preprocessing and data collection. Guojun PENG and Yining LI contribute throughout all stages of model development and portfolio construction. On a different front, Jiamu CHEN is primarily in charge of drafting the paper and conducting an in-depth investigation of the 'SWIFT' methodology, including frameworks, settings, and parameters.

8 References

References

[1] Zhao, J., Wang, T., Abid, W., Angus, G., Garg, A., Kinnison, J., Sherstinsky, A., Molino, P., Addair, T., & Rishi, D. (2024). *LoRA Land: 310 Fine-tuned LLMs that Rival GPT-4, A Technical Report.*

[2] Iacovides, G., Konstantinidis, T., Xu, M., & Mandic, D. (2024). *FinLLaMA: LLM-Based Financial Sentiment Analysis for Algorithmic Trading.*

[3] Chiu, I.-C., & Hung, M.-W. (2024). Finance-specific large language models: Advancing sentiment analysis and return prediction with LLaMA 2.

[4] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.*

[5] Li, Yixiao, et al. "Loftq: Lora-fine-tuning-aware quantization for large language models." arXiv preprint arXiv:2310.08659 (2023).

[6] Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." Advances in neural information processing systems 36 (2023): 10088-10115.

[7] Zhao, Yuze, et al. "Swift: a scalable lightweight infrastructure for fine-tuning." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 39. No. 28. 2025.

[8] Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063.*

[9] Chen, Yifei, Bryan T. Kelly, and Dacheng Xiu. "Expected returns and large language models." Available at SSRN 4416687 (2022).

[10] Kirtac, K., & Germano, G. (2024). Sentiment trading with large language models. *Finance Research Letters*, 62(B):105227.

[11] Huang, A. H., Wang, H., & Yang, Y. (2023). FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2), 806–841.