Efficient Fine-Tuning of Financial Language Models with QLoRA for Alpha Signal Generation

Fusheng Luo[1], Guojun Peng[1], Jiamu Chen[1], Yining Li[1]
[1] Johns Hopkins University, USA
fluo5@jh.edu, gpeng9@jh.edu, jchen471@jh.edu, yli688@jh.edu

Abstract

In this study, we investigated the effectiveness of QLoRA-based fine-tuning on Meta-LLaMA3-8B for the classification of financial sentiment and signal generation. Leveraging a diverse set of domain-specific datasets and the SWIFT framework, our model achieved superior performance over established baselines—such as FinBERT (a classic sentiment analyzer in academia), RoBERTa, and VADER—in both sentiment classification and downstream trading strategies on a separate financial news dataset.

Keywords: Financial Sentiment Analysis, Large Language Models, QLoRA, LLaMA, Fine-tuning, Market-Neutral Trading Strategy, K-means Clustering

1 Introduction

Large Language Models (LLMs) have become increasingly popular in financial text analysis, providing notable advancements over traditional dictionary-based approaches in extracting predictive insights from unstructured sources like earnings calls, news reports, and regulatory documents. Their advantage is in capturing contextual details and sentiment subtleties often overlooked by simpler methods. However, generic LLMs encounter two main challenges in finance: (1) a lack of domain-specific training, which can cause misunderstandings of financial terminology, and (2) the high computational costs

associated with full-model fine-tuning, making them less accessible for many users.

Recent studies have investigated fine-tuning foundation models like LLaMA with finance-focused data to enhance return predictions. While optimizing the entire model yields high performance, it demands substantial resources. Parameter-efficient methods such as Low-Rank Adaptation (LoRA) provide a viable alternative by adjusting only a limited number of parameters, thus lowering memory use. However, there is still a balance to strike between tuning efficiency and accuracy, particularly for users with hardware limitations.

This study explores QLoRA, a fine-tuning method combining 4-bit quantization and LoRA for efficient, scalable training without performance loss. QLoRA keeps the base model weights fixed in 4-bit precision and adds trainable 16-bit LoRA adapters, enabling models up to 65 billion parameters to be fine-tuned on a single 48GB GPU [Dettmers et al., 2023] QLoRA has shown competitive results in general NLP tasks; its use in financial modeling remains largely unexplored. This work assesses whether QLoRA can preserve strong predictive capabilities for alpha signal generation based on financial news sentiment.

Beyond the financial domain, Zhao et al. [Zhao et al., 2024] evaluated LoRA-based fine-tuning across 310 LLMs, covering 31 tasks and 10 base models. Their findings reveal that 4-bit LoRA models frequently outperform their base models and even GPT-4 on certain tasks, while utilizing considerably less memory. They also introduce LoRAX, an inference server able to run multiple LoRA adapters simultaneously on a single GPU, demonstrating the practicality of efficient parameter deployment.

In finance, FinLLama [Iacovides et al., 2024] customizes LLaMA-2 using LoRA and 8-bit quantization on labeled financial sentiment data. It efficiently generates sentiment scores and trading signals while reducing computational demands [Chiu and Hung, 2024]. Nonetheless, FinLLama does not explore finer quantization levels or lighter infrastructure options. Conversely, QLoRA addresses these limitations by incorporating NormalFloat 4-bit quantization, double quantization, and paged optimizers, enabling full 16-bit training quality on low-resource hardware. Models like Guanaco, developed within this framework, compete with ChatGPT on benchmarks such as Vicuna [Dettmers et al., 2023], demonstrating QLoRA's potential for accessible, high-performance fine-tuning in finance.

To clearly illustrate our workflow, Figure 1 provides an overview of the end-

to-end pipeline for our sentiment-driven alpha signal generation system. We begin with data collection from four financial sentiment datasets—Financial PhraseBank, NASDAQ News Sentiment, Twitter Financial News, and FIQA2018—which are then preprocessed via label normalization and tokenization using the LLaMA3 tokenizer. The resulting data is used to fine-tune the Meta-LLaMA3-8B model using QLoRA, a 4-bit quantized parameter-efficient method. The fine-tuning process is orchestrated via the SWIFT framework, which facilitates scalable and hardware-efficient training. The fine-tuned model is then applied to financial news headlines, producing discrete sentiment scores that are aggregated per stock. Finally, we design and evaluate three market-neutral trading strategies based on the predicted sentiments. Across all setups, our approach demonstrates superior performance compared to both the S&P 500 benchmark and alternative sentiment-based models, highlighting its practical value in quantitative trading.

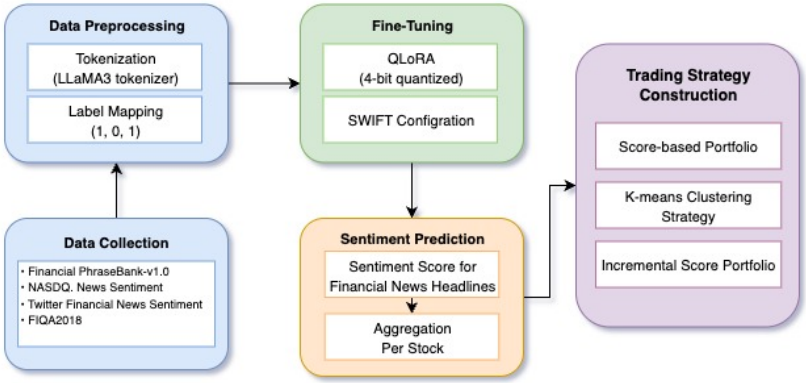

Figure 1: Workflow pipeline for financial sentiment modeling and trading.

1.1 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:

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

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.

2 Data Sources and preprocessing

2.1 Dataset

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.

Part 1: Fine-tuning Dataset We generally use four datasets here

(i) Financial PhraseBank-v1.0

(ii) NASDQ news sentiment

(iii) Twitter Financial News Sentiment

(iv) 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.

Part 2: Predicting Real-Time News We use two datasets: the news dataset, which includes news articles and alerts (with alerts representing news with only headlines), and the corresponding stock price dataset. We collected news text data from Refinitiv (now known as LESG), CRSP, and Yahoo Finance for stock prices. Then, we create a news dataset covering a list of stocks within a time range of 16 months (2024.01.01 - 2025.05.30). This news dataset includes 502 stocks [S&P Dow Jones Indices, 2025] and 77000 news headlines. Additionally, for these 502 companies, we also have stock prices and volume data covering the same period from Yahoo Finance. This is the fundamental dataset we use to evaluate our trading strategies in comparison to the benchmark strategies discussed in the following sections.

2.2 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.

2.2.1 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.

2.2.2 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.

3 Methodology

3.1 QLoRA

LoRA Adaptation. Low-Rank Adaptation (LoRA) [Hu et al., 2022] is a parameter-efficient fine-tuning method that freezes the original pretrained weight matrix $W_0 \in R^{m\times n}$ and introduces two small trainable matrices $A \in R^{m\times r}$ and $B \in R^{r\times n}$, where $r \ll n$. The weight update is parameterized as

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

so that the forward computation during fine-tuning becomes

$$y = (W_0 + \tfrac{\alpha}{r} AB)x, \tag{2}$$

where α is a scaling factor controlling the contribution of the LoRA update.

In our implementation, we set $r = 8$ and $\alpha = 32$, applying adapters to all linear layers (including attention and feed-forward projections). This configuration introduces less than 0.05% additional trainable parameters while maintaining full model expressiveness.

Quantile. Using the LoRA components described above, we define QLoRA [Dettmers et al., 2023] for a single linear layer in the quantized base model with a LoRA adapter as follows:

$$Y^{\mathrm{BF16}} = X^{\mathrm{BF16}} \cdot \mathrm{doubleDequant}\big(c_1^{\mathrm{FP32}},\, c_2^{\mathrm{FP8}},\, W^{\mathrm{NF4}}\big) + X^{\mathrm{BF16}} \cdot \frac{\alpha}{r} A^{\mathrm{BF16}} B^{\mathrm{BF16}}. \tag{3}$$

where $\mathrm{doubleDequant}(\cdot)$ reconstructs the quantized weight matrix from its two-level quantization scheme:

$$\mathrm{doubleDequant}(c_1, c_2, W) = \mathrm{dequant}\big(\mathrm{dequant}(c_1, c_2, W)\big) = W^{\mathrm{BF16}}. \tag{4}$$

In our configuration, we use the `NF4` quantization format for W, which provides a non-uniform 4-bit representation optimized for model performance, and we apply *double quantization* to compress the scaling factors c_1 and c_2 into low-bit formats. All matrix multiplications and LoRA updates are performed in `bfloat16` precision to ensure numerical stability and efficient utilization of modern GPUs.

During training, only the LoRA adapter parameters A and B are updated, while the quantized backbone weights remain frozen. The gradients with respect to the LoRA parameters are computed using the reconstructed W^{BF16} obtained from the double dequantization process. This design significantly reduces memory consumption while maintaining training stability and model quality.

3.2 Swift

SWIFT (Scalable lightWeight Infrastructure for Fine-Tuning) is an open-source [Zhao et al., 2025], unified framework designed to support the full lifecycle of Large Language Models (LLMs) including training, fine-tuning, evaluation, and deployment. A key feature of SWIFT is its modular and extensible architecture, which enables users to train large-scale models efficiently on limited hardware.

To facilitate efficient fine-tuning of the LLaMA 3 model, we incorporate components from the SWIFT framework. While our overall pipeline is based on Transformers, we specifically use SWIFT's "Swift" and "TrainingArguments" modules to simplify configuration and the integration of parameter-efficient adapters. These modules guarantee reliable and streamlined LoRA integration and training setup for complex models, especially in 4-bit quantized form, without needing manual patching or specifying target modules.

3.3 Benchmark Comparison

We benchmark our QLoRA-fine-tuned LLaMA3-8B model against a suite of established financial sentiment analysis models, including FinBERT, VADER, and Financial RoBERTa, to assess performance across both classification accuracy and downstream trading signal generation. All models are evaluated on the same dataset to ensure consistency (our own test/out-of-sample dataset).

FinBERT is a BERT-based encoder-only model developed by ProsusAI [Araci, 2019], fine-tuned specifically for sentiment classification tasks in the financial domain. It was trained on a subset of the Financial PhraseBank, which contains manually annotated sentences from financial news. The model outputs a softmax distribution over three sentiment classes—positive, neutral, and negative—and is widely used for traditional text classification pipelines.

VADER (Valence Aware Dictionary and sEntiment Reasoner) [Hutto and Gilbert, 2014] is a rule-based sentiment analysis tool optimized for social media and short financial text. Unlike transformer-based models, VADER relies on a lexicon of sentiment-labeled words and empirically tuned heuristics to produce a compound sentiment score ranging from –1 (most negative) to +1 (most positive). For benchmarking purposes, we discretize this continuous output into three sentiment categories using empirically chosen thresholds.

Financial RoBERTa, developed by Soleimani et al. [Soleimanian, 2023], is a regression-based RoBERTa variant trained on financial text to predict a continuous sentiment score in the range of –1 to +1. It is fine-tuned on curated financial datasets using a mean-squared-error objective and demonstrates strong correlation with market-relevant sentiment. Like VADER, its output is also discretized into three sentiment classes for comparison with other models.

This diverse set of models—spanning dictionary-based, regression-based, classification-based, and generative paradigms—allows for a robust benchmarking of our QLoRA-LLaMA3 model's generalization and domain understanding capabilities.

3.4 Trading Strategy

In this section, we focus on generating market-neutral strategies. Such strategies isolate the alpha from the beta, and any subsequent returns captured are attributed to the model's skills rather than the market movements. Market-neutral strategies are also more robust under stress test and regulatory scrutiny. We also maintain a dollar-neutral position, which means that our methods generally long-short the same amount of money (USD) to lower the systematic risk (portfolio beta is near zero). From these stances, we continue to build two trading mechanisms.

3.4.1 Score-based portfolio

To assess the practical profitability of our sentiment classification model, we have developed a straightforward portfolio strategy based on daily news sentiment predictions [Chen et al., 2022]: score-based portfolio. For each stock, we aggregate all predicted sentiment labels from the daily news to compute an average sentiment score. Utilizing this score, we create a zero-net-investment

portfolio by taking equal-weighted long positions in the top quintile (15%) of stocks exhibiting the most positive sentiment, while simultaneously shorting the bottom quintile (15%) with the most negative sentiment. The portfolio is not rebalanced daily, as news coverage for all stocks may not be available every day. To address this issue, we establish a minimum threshold of 30 for the number of sentiment scores generated. If the total number of sentiment scores produced in a day exceeds this threshold, we consider that a trading day is effective for reconstructing our trading portfolio. This approach allows us to adjust our positions based on market conditions rather than making arbitrary changes, which could lead to increased explicit and implicit trading costs.

3.4.2 K-means clustering trading strategy

We design a market-neutral, static long-short trading strategy by clustering stocks based solely on their time-series sentiment patterns, without using any price or return information. Specifically, we construct a matrix of our-model's sentiment scores with tickers as rows and dates as columns, then apply KMeans clustering across the entire time period. This groups stocks into clusters based on how their sentiment evolves over time. We then compute an aggregate sentiment score for each ticker (e.g., average or cumulative sentiment), and rank entire clusters based on the average sentiment of their members.

From this ranking, we long the tickers in the top 20% of clusters (those with the most positive sentiment trajectories) and short the tickers in the bottom 20% of clusters (those with the most negative sentiment). These long and short positions are held unchanged throughout the entire period (no rebalancing). In particular, since the number of tickers may differ for the top and bottom clusters, we just long and short $ 1 total for each position, i.e. if we have 120 tickers for short position, we only sell $ 1, the same happens for the long position. Portfolio performance is evaluated by computing the daily spread return (long minus short), accumulating it over time, and analyzing metrics such as Sharpe ratio, max drawdown, and annualized volatility — all while strictly avoiding look-ahead bias.

3.4.3 Incremental score-based portfolio

Furthermore, [Kim et al., 2023] suggests that the (nearly) daily news can often be noisy and unreliable for constructing our trading strategies. To address this issue and in line with their analysis, we propose an incremental score-based portfolio. The concept revolves around measuring changes in sentiment scores between two consecutive effective trading days, allowing us to capture the "spike" in market sentiment. To hit this level of thinking, we forward fill all the null values in the sentiments column, reflecting the unchanged market emotions towards any particular stock. This approach helps us distinguish between noise and genuine market trends. Our ranking method is straightforward; we subtract the sentiment score of a particular stock on trading day i from that of trading day $i-1$. We then rank these increments by their values and select the top 20% for long positions and the bottom 20% for short positions.

Additionally, we account for the lag in the societal impact of news; therefore, we base our position on the news from yesterday and make trades today. We will hold this position for 2 days to capture the directional return from our factors.

3.4.4 Benchmark trading strategy

As the cornerstone of our trading strategy evaluation, we adopt a long-only investment approach in the S&P 500 index as our benchmark. This benchmark serves as a representative proxy for broad market performance, enabling us to assess the relative value added by our sentiment-driven model. By comparing against a passive, well-diversified index strategy, we establish a robust baseline to evaluate whether our active strategy generates excess return (alpha) beyond general market trends.

Algorithm 1 Score-Based Portfolio Construction

for each day t **do**
    Extract all sentiment signals: `sent[day t]`
    if number of valid (non-NaN) scores ≥ 30 **then**
        Sort stocks by sentiment score
        Select:

- Top 20% $\rightarrow$ long \$1
- Bottom 20% $\rightarrow$ short \$1

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

Algorithm 2 Incremental Score-Based Portfolio Construction

for each trading day t **do**
    if both t and $t-1$ are effective trading days (i.e., sentiment signals ≥ 30) **then**
        Forward-fill missing sentiment values up to day t
        Extract sentiment scores: `sent[day` t`]` and `sent[day` $t-1$`]`
        Compute sentiment increment: `delta = sent[day` t`] - sent[day` $t-1$`]`
        Sort stocks by `delta`
        Select:

- Top 20% $\rightarrow$ long 5% of capital
- Bottom 20% $\rightarrow$ short 5% of capital

        Execute positions based on sentiment from day $t-1$ on day t
        Hold positions for 2 trading days
        Calculate P&L for the 2-day holding period
    end if
end for

4 Experiments and Results

4.1 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.

4.1.1 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.

4.2 Evaluation Matices

4.2.1 Classification evaluation metrices

We use the micro-level F1 score, precision, recall to measure the model's predictability on a given class. As the model varies, the sentiment dataset may be highly imbalanced, this is where we introduce the macro-level average and weighted average indices in the classification reports in tables 1 and 2.

4.2.2 Portfolio's performance evaluation metrices

How do we evaluate the performance of our trading strategies? For our market-neutral approaches, it is useful to compare metrics such as the Sharpe ratio, maximum drawdown, annualized return, and volatility. Since we generate directional factors, adding the hit ratio as an evaluation metric is also beneficial. We rebalance our long and short positions daily. If a long position gains or a short position (including reversed positions) profits, it counts as a "hit"; otherwise, it's a "miss." We then compute the ratio of hits to total hits and misses. This provides a clear measure of the model's ability to predict market directions.

4.3 Results

4.3.1 Models' performances on sentiments classification tasks

We evaluated our fine-tuned Meta-LLaMA3-8B sentiment classifier on a test set of 9,064 financial news samples. From table 1, 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.

Table 1: Classification Performance of Our QLoRA-tuned LLaMA3-8B Model

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Negative	0.14	0.03	0.06	144
Neutral	0.61	0.88	0.72	983
Positive	0.99	0.94	0.96	7937
Accuracy			0.92	
Macro Avg	0.58	0.62	0.58	9064
Weighted Avg	0.93	0.92	0.92	9064

For baseline comparison (in Table 2), we evaluated FinBERT [**?**] on the same test set. Although FinBERT achieved high precision for the Positive class (0.8232), its overall recall was significantly lower, resulting in a $F1$ score of

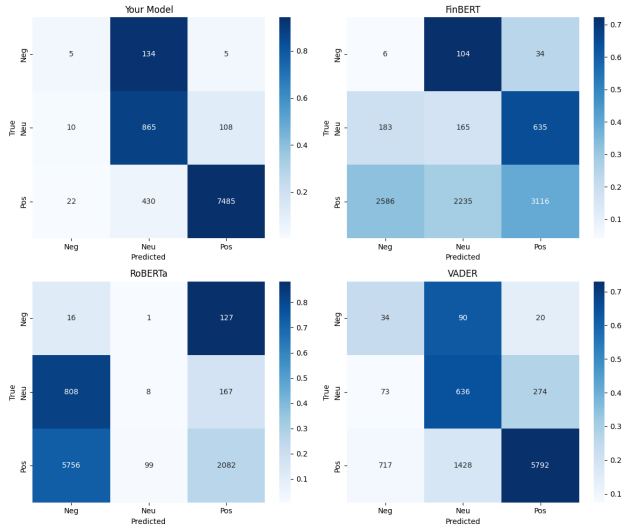

Figure 2: Confusion Matrices of Sentiment Classifier.

only 0.5316 for that class. Performance in the negative and neutral classes was notably poor, with an overall accuracy of just 36.26% and a macro $F1$ of 0.2101.

For baseline comparison, we evaluated the rule-based sentiment analyzer VADER [Hutto and Gilbert, 2014] on the same test set. VADER achieved strong precision on the Positive class (0.9466), but significantly underperformed on the Negative class ($F1$ score of only 0.0659) and the Neutral class ($F1$ of 0.4101). The model exhibited a tendency to overpredict Positive sentiment, resulting in an imbalanced performance. Overall accuracy was 71.05%, while macro $F1$ remained modest at 0.4303.

We also evaluated the Financial RoBERTa model [Soleimanian, 2023], which outputs continuous sentiment scores subsequently discretized into three classes. Although the model showed high precision for the Positive class (0.8800), its recall was relatively poor (0.2579), leading to an $F1$ score of 0.3991 for that class. Performance on the Negative and Neutral classes was notably weak, with near-zero F1 scores. The overall accuracy reached just 23.11%, and macro $F1$ dropped to 0.1436, suggesting poor generalization beyond majority sentiment prediction.

Additionally, we schematically present the confusion matrices of the four models in Figure 2. From the figure, it is evident that our LLaMA3-based classifier substantially outperforms the baselines across all three sentiment classes. Notably, our model achieves strong performance in the neutral

Table 2: Performance Comparison of Baseline Models (without Support column)

Model	Class	Precision	Recall	F1-Score
VADER	Negative	0.04	0.24	0.07
	Neutral	0.30	0.65	0.41
	Positive	0.95	0.73	0.83
	Accuracy		0.71	
	Macro Avg	0.43	0.54	0.43
	Weighted Avg	0.87	0.71	0.77
FinBERT	Negative	0.00	0.04	0.00
	Neutral	0.07	0.17	0.09
	Positive	0.82	0.39	0.53
	Accuracy		0.36	
	Macro Avg	0.30	0.20	0.21
	Weighted Avg	0.73	0.36	0.48
RoBERTa	Negative	0.00	0.11	0.00
	Neutral	0.07	0.01	0.01
	Positive	0.88	0.26	0.40
	Accuracy		0.23	
	Macro Avg	0.32	0.13	0.14
	Weighted Avg	0.78	0.23	0.36

class, with minimal confusion between neutral and positive instances, a challenge that significantly impacts FinBERT and RoBERTa. FinBERT shows a strong bias toward predicting the positive class, with over 2,500 Negative and 600+ Neutral examples misclassified as Positive. Similarly, RoBERTa heavily overpredicts the positive class while failing to recognize Neutral and Negative sentiments. VADER performs comparatively better than FinBERT and RoBERTa in distinguishing Neutral from Positive, but still exhibits class imbalance and reduced granularity. Although the Negative class remains underrepresented and challenging for all models, our fine-tuned LLaMA3 classifier demonstrates higher robustness and generalization, making it a reliable choice for downstream financial sentiment analysis.

4.3.2 Models' performances on S&P 500 companies' news

Histogram of news sentiments Following Figure 3, we can have a glimpse of the general sentiment prediction distribution for all four models.

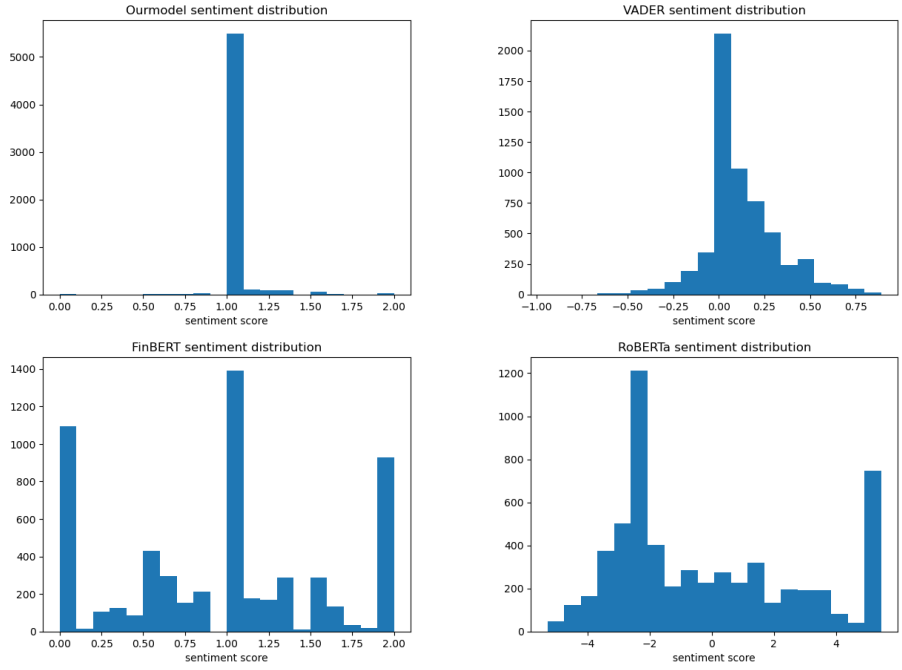

Figure 3: Histograms of sentiments for four different models.

Compared to the other three models, our model generally adopts a more conservative approach, typically predicting near-neutral sentiment. In contrast, FinBERT tends to forecast more extreme sentiments more aggressively. RoBERTa shows a tendency toward positive predictions but mostly assigns negative sentiments to most news. The models' performances vary across different news datasets. Although our dataset is limited to only 15 months, this isn't enough to definitively determine whether our model is better or worse than the others. Additionally, we evaluate how these sentiments influence trading outcomes.

Performance on score-based trading strategies As Figure 4 and Table 3 show, the VADER and ourmodel in this score-based method outperform the benchmark SPY almost over the whole period (except for 2024-11 to 2025-03), achieving higher cumulative returns with relatively stable trajectories and resulting in higher return at the end (2025-05). In particular,

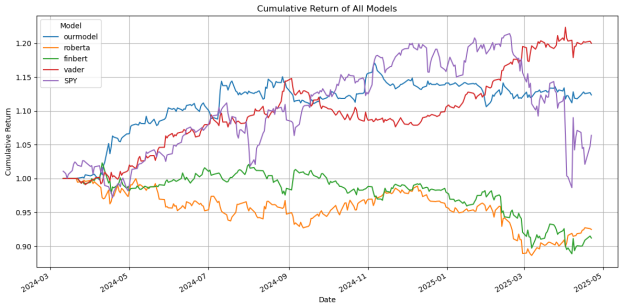

Figure 4: Cumulative return of the score-based trading strategy.

Table 3: Performance Metrics of Sentiment-Based Trading Models (Score-based)

Metric	**OurModel**	**Roberta**	**FinBERT**	**VADER**
Annualized Return	11.06%	-6.80%	-7.95%	17.84%
Annualized Volatility	8.28%	9.67%	9.33%	8.58%
Sharpe Ratio	1.31	-0.68	-0.84	1.96
Max Drawdown	-5.50%	-11.37%	-13.14%	-6.23%
Hit Ratio	49.45%	48.71%	49.26%	50.00%

our model demonstrates a stable profit accumulation, yielding an annualized return of more than 11%, a Sharpe ratio of over 1.3, while processing a relatively low annualized volatility of 8.28% and a maximum drawdown of -5.50%, compared to all other models. This demonstrates the profitability of our model within this trading strategy, even though it predicts in a relatively conservative way. We believe that this model's backward-looking sentiment predictions would greatly benefit a more hype-affluent market sector.

On the other hand, VADER performs exceptionally well, even better than our model, with a Sharpe ratio of more than 1.90. We believe its profitability can be attributed to its relatively conservative forecasting, and its fatter tail captures more market momentum than our model. However, in the subsequent trading scheme, you will find that this model ultimately proves less prosperous.

Performance on incremental score-based trading strategies All four models perform in a highly similar pattern in this trading strategy. As figure 4 verifies our incremental-score mechanism has filtered out the majority amount of noise in the forecasts. While OurModel demonstrated robust

Table 4: Performance Metrics of Sentiment-Based Models with 2-Day Holding Period, Same-Day Entry

Metric	**OurModel**	**Roberta**	**FinBERT**	**VADER**
Annualized Return	4.17%	2.29%	2.64%	2.48%
Annualized Volatility	4.57%	4.83%	4.60%	4.51%
Sharpe Ratio	0.92	0.49	0.59	0.57
Max Drawdown	-4.51%	-4.07%	-3.25%	-4.12%
Hit Ratio	49.63%	49.81%	50.93%	50.00%

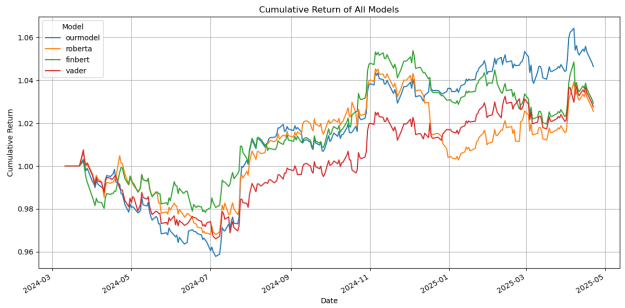

Figure 5: Cumulative return of the incremental score-based trading strategy.

consistency and risk-adjusted performance across various trading regimes, it showed limitations in maximizing raw returns. In particular, it was often outperformed by VADER in terms of absolute annualized return, suggesting its signals may be more conservative and less aggressive in capturing strong directional moves. Additionally, its hit ratio hovered around 49–50%, indicating that the model's predictive accuracy is only marginally better than random chance, relying heavily on position sizing and risk controls to generate profit.

This modest predictive power could present challenges if market volatility or trend patterns change, potentially exposing the model to suboptimal trade entries or exits in less stable environments. Further improvements in signal quality, for example through feature engineering or ensemble calibration, could help OurModel increase its raw returns without compromising its strong drawdown control.

Performance on K-means clustering (cluster number = 7) trading strategies As shown in Figure 6, the trading strategy derived from our model outperforms the one based on FinBERT when using sentiments

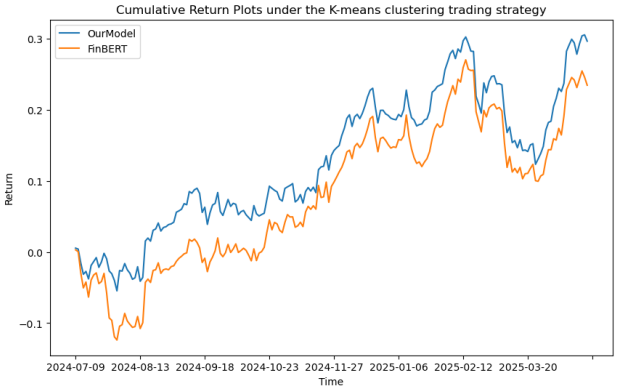

Figure 6: Cumulative return of the incremental score-based trading strategy.

generated by our approach. Specifically, our model achieves a Sharpe ratio of 1.686, compared to 1.280 for FinBERT. Both models have similar maximum drawdowns, around 0.17. While our model clusters 400 tickers into the bottom portfolio, FinBERT only uses 14 tickers. Nonetheless, our approach yields higher profits on the same news dataset.

5 Conclusion

In this project, we investigated how parameter-efficient fine-tuning techniques can be applied to financial sentiment analysis, focusing on Quantized LoRA (QLoRA) with the Meta-LLaMA3-8B model. By combining multiple financial sentiment datasets and utilizing the SWIFT framework, we trained a domain-specific model capable of producing high-quality sentiment signals from financial news headlines.

Our results indicate that the QLoRA-tuned LLaMA3-8B model achieves over 92% accuracy in classification, with strong precision and recall for neutral and positive sentiments. In trading scenarios, it showed consistent profitability and favorable risk-adjusted returns, outperforming traditional models in metrics like the Sharpe ratio and maximum drawdown under score-based strategies. Even in the incremental score-based and K-means clustering approach, our model remained competitive, though its conservative nature slightly limited raw returns compared to VADER.

These findings highlight the promise of quantized, parameter-efficient fine-tuning of large language models for financial tasks—striking a good balance

between performance, efficiency, and ease of deployment. Future work could improve this framework by adding continuous sentiment scoring, integrating multi-modal inputs such as charts or analyst transcripts, and applying reinforcement learning to directly optimize trading signals. Exploring ensemble techniques and better handling of class imbalance, especially for negative sentiments, could also boost the model's robustness.

Looking ahead, extending this approach to generate continuous sentiment scores instead of discrete classifications could offer more nuanced signals. Incorporating broader market context features and leveraging reinforcement learning to directly optimize trading strategies are promising avenues for enhancing both prediction accuracy and economic gains.

Ethics statement

This study did not involve human participants, animals, or sensitive data. Therefore, no ethical approval was required.

Author contributions

FSL devoted a majority of work in this paper. He designed the study, wrote the manuscript, and lead the direction of researches with PGJ; the LLM is trained by FSL with the help of CJM, who contributes to the detailed study of SWIFT packages.

All authors reviewed and approved the final manuscript.

Funding

This research received no external funding. The authors did not receive financial support from any public, commercial, or non-profit funding agencies.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgments

The authors thank Meta for releasing the LLaMA model as open-source software. No financial support, sponsorship, or research funding was received from Meta or any other organization.

Data availability statement

The datasets and code used in this study are available at: `https://github.com/RomeoisFushengLuo/TradingStrategies_NewsSentimentAnalysis_FinLlama3_8B`.

References

[Araci, 2019] Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models.

[Chen et al., 2022] Chen, Y., Kelly, B. T., and Xiu, D. (2022). Expected returns and large language models. Available at SSRN 4416687.

[Chiu and Hung, 2024] Chiu, I.-C. and Hung, M.-W. (2024). Finance-specific large language models: Advancing sentiment analysis and return prediction with llama 2. *arXiv preprint arXiv:2402.08887*.

[Dettmers et al., 2023] Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

[Hu et al., 2022] Hu, E. J. et al. (2022). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[Hutto and Gilbert, 2014] Hutto, C. J. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM)*, Ann Arbor, MI.

[Iacovides et al., 2024] Iacovides, G., Konstantinidis, T., Xu, M., and Mandic, D. (2024). Finllama: Llm-based financial sentiment analysis for algorithmic trading. *arXiv preprint arXiv:2401.07874*.

[Kim et al., 2023] Kim, S., Kim, S., Kim, Y., Park, J., Kim, S., Kim, M., Sung, C. H., Hong, J., and Lee, Y. (2023). Llms analyzing the analysts: Do bert and gpt extract more value from financial analyst reports? In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 383–391.

[Soleimanian, 2023] Soleimanian, M. (2023). Financial roberta: A roberta-based language model fine-tuned for financial sentiment analysis. Accessed: 2025-06-19.

[S&P Dow Jones Indices, 2025] S&P Dow Jones Indices (2025). S&p 500 index. Accessed: 2025-06-25.

[Zhao et al., 2024] Zhao, J., Wang, T., Abid, W., Angus, G., Garg, A., Kinnison, J., Sherstinsky, A., Molino, P., Addair, T., and Rishi, D. (2024). Lora land: 310 fine-tuned llms that rival gpt-4: A technical report. *arXiv preprint arXiv:2406.XXXXX*.

[Zhao et al., 2025] Zhao, Y. et al. (2025). Swift: A scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39.