First Come First Serve: Outpacing the Market with Automated Financial News Sentiment Analysis.

Team Members:
Aksel Joonas Reedi (s4790820) - a.j.reedi@student.rug.nl
Łukasz Sawala (s5173019) - l.h.sawala@student.rug.nl
Mika Umaña Lemus (s5173213) - m.e.umana.lemus@student.rug.nl

GitHub Repository:
`https://github.com/akseljoonas/biotech-news-sentiment`

January 18, 2025

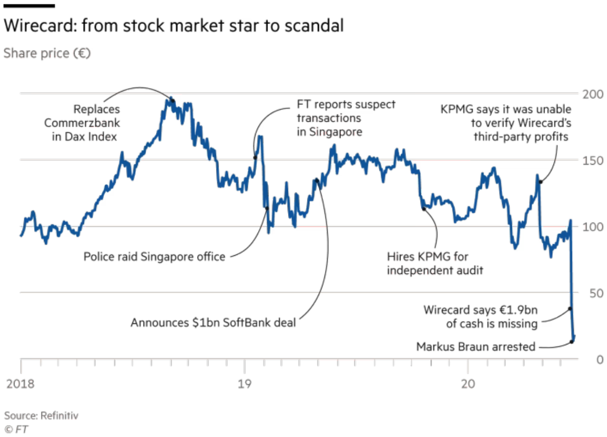
1 Abstract

This paper explores the application of Large Language Models (LLMs) for automated sentiment analysis of financial news to predict stock market movements, specifically focusing on the biotech sector. We compare various BERT-based models, including domain-specific variants like FinBERT and BioBERT, evaluating their performance using both traditional F1 scores and a custom profit metric that simulates real-world trading decisions. Our research utilizes a novel dataset of 4,069 press releases from 118 biotech companies, with price movements tracked from one minute before to eight hours after each announcement. The results demonstrate that domain-specific pre-training (FinBERT) significantly outperforms superior architectures (DeBERTa) in financial applications, achieving a 168% profit metric despite a modest F1 score of 0.576. We also investigate parameter-efficient fine-tuning methods using Low-Rank Adaptation (LoRA), which maintains reasonable performance while reducing computational requirements. Our findings suggest that combining domain-specific pre-training with innovative fine-tuning methods can create models that reliably generate profits while outpacing market competitors. Additionally, we highlight the importance of domain-specific evaluation metrics, as traditional classification metrics don't necessarily correlate with real-world trading performance.

2 Introduction

In the U.S., 62% of adults own various investments such as stocks, treasury bonds, or commodities. Information about public companies—such as earnings reports, product launches, and financial metrics—is widely accessible online.

Although these updates are often written in a neutral tone, they often contain analysts' opinions or other implicit sentiments that can be effectively analyzed using sentiment analysis. This allows for a more nuanced understanding of market expectations. Furthermore, these sentiments are often closely linked to short-term stock price movements, as seen in an example of Wirecard stock movement after major press releases in Figure 1.


Wirecard: from stock market star to scandal

Share price (€)

Source: Refinitiv
© FT

Figure 1: Timeline of Wirecard's stock price movements in response to major press releases, demonstrating the significant impact of news sentiment on market behavior. The graph shows sharp price declines following negative news releases, particularly during the company's accounting scandal, illustrating the direct relationship between news sentiment and stock price movements.

Predicting those stock movements is a complex task, usually requiring vast analytical manpower and multiple domain experts to react quickly and make the right decision to benefit from the resulting market movement.

Since the rise of digitalized trading, multiple AI stock market products have been developed and deployed, with one of the most famous examples being the Flash Crash[1], where the misuse of automated trading algorithms caused the whole US market to drop by 9% in a few minutes. This example among multiple others dropped most research on automated traders based on pure, statistical reasoning.

Recent advances in Natural Language Processing (NLP), particularly with Transformer-based Large Language Models (LLMs) (Vaswani et al., 2023), have transformed this field. Instead of designing systems to outperform humans in stock trading, researchers have shifted their focus to replicating top investors' strategies(Guo & Hauptmann, 2024) to maximize profits. This shift has been driven by the exceptional capacity of these models to handle and retain vast amounts of data(Devlin et al., 2019). Their ability to excel in tasks like text summarization and sentiment analysis, which were challenging for earlier architectures, has opened new possibilities in the financial domain.

Sentiment analysis (SA) is a task in the field of NLP that consists of determining and classifying the

[1] https://en.wikipedia.org/wiki/2010_flash_crash

tone (sentiment) of a text in a particular context. The ability to detect the contextual meaning of a text is crucial as it allows one to extract and separate emotions, opinions, and attitudes from the direct wording of any text. Therefore, SA could be used to reliably analyze whether a press release and the underlying announcement could influence the stock price by a significant margin.

This project leverages LLMs to perform automated sentiment analysis, aiming to anticipate market reactions to a particular news piece faster than competitors such as investment banks. Predicting market responses to news seeks to financially benefit the users of the model developed in this research through strategic buying or shorting of stocks, effectively outpacing the rest of the market.

This research idea is not novel, with multiple examples of papers touching upon this topic from multiple angles, either through the development of SA financial datasets such as Financial PhraseBank (Malo et al., 2013), developing new foundational models such as FinBERT(Araci, 2019) or by attempting automated trading in general (Ding et al., 2024). However, the team has not found evidence of research that combines all those findings into an end-to-end LLM trading application with a customizable dataset, multiple training methods, and custom profit metrics to ensure the quality of the whole process, which is what this research aims to do.

To conclude this section, it is important to emphasize that stock trading is typically framed as a time series regression problem, which is inherently more complex and susceptible to noise. As suggested by the team's domain expert(mentioned at the end of the report), approaching it as a Sentiment Analysis problem may offer notable improvements in model performance. This happens as traders often base their trading decisions on sentiment—such as increased interest in purchasing stocks when positive financial results are reported. By focusing on sentiment-driven signals, the task becomes more structured, potentially leading to more accurate predictions.

3 Related Work

Sentiment analysis is a multi-class classification task, requiring a deep understanding of the context and language semantics combined with high emotional intelligence. This led to a focus on encoder-based LLMs in this research domain, as their bidirectional attention mechanism could capture much more information than their counterparts used in the decoder architectures. However, this paradigm of exclusively using encoder-based models has been recently challenged, with decoder-based architectures also beginning to show promise. This shift will be explored further in the Future Research section. In general, recent advancements in LLMs have significantly enhanced sentiment analysis performance but also raised the need for more efficient training methods to handle the large computational demands associated with fine-tuning such models.

3.1 Enhanced Training Techniques

Several strategies have been developed to improve model performance during fine-tuning while also addressing the computational complexity involved in training LLMs.

Gradual unfreezing (Howard & Ruder, 2018) is one the most efficient methods as it helps to prevent catastrophic forgetting and significantly reduces training time by freezing most of the parameters initially, gradually increasing the set of trainable parameters throughout training. This idea stems from the observation that most task-specific knowledge seems to be encoded in the higher layers of the models, hence there is little to no benefit from changing the initial layers, which also minimizes the risk of catastrophic forgetting(Chen et al., 2020).

Another widely-used method is the slanted triangular learning rate (SLTR) (Howard & Ruder, 2018) which is a framework that defines the schedule of the learning rate values over time. Initially, the learning rate is increased (the "warm-up" period). After the learning rate reaches a predefined peak the learning rate drops off again. This process ensures stability over the whole learning process, allowing the model to gradually learn the task instead of using a constant learning rate.

The most commonly used optimizer in ML and LLMs is ADAM (Adaptive Moment Estimation)(Kingma & Ba, 2017). ADAM combines momentum and adaptive learning rates in order to exit local minima and minimize the noise of the updates by keeping track of the few previous gradient steps. Applying ADAM is helpful in most NN applications, but becomes almost necessary when applying full fine-tuning.

Those three techniques are considered to be one of the main building blocks of any successful complex LLM application. To assess their necessity for this research, the team explored state-of-the-art evaluation methods to determine their relevance and effectiveness for the proposed model.

3.2 Custom Evaluation Metrics

NLP research has highlighted the use of alternative evaluation metrics that give more information on the actual magnitude of mistakes made by the model (Du et al., 2024). In the financial domain, it is particularly important to focus on the investment decisions that significantly affect performance, meaning that the impact of an error can vary depending on the context. This evaluation trait does not always align with the best-performing model according to standard metrics. Conventional evaluation methods often fail to capture this critical aspect of model performance, which is essential for applications like financial predictions. Therefore, implementing custom metric tailored to the needs of the project is crucial to properly evaluate model perfomance.

3.3 Efficient Fine-tuning Methods

Due to the large number of parameters and enormous dataset sizes, countless efficiency methods have been introduced so that training can be done while using as few resources as possible without heavily impacting final model performance. In the field of LLMs, standard ML techniques such as early stopping with patience or simulating batching with gradient accumulation are expanded with so-called Parameter-efficient Fine-Tuning (PEFT) methods. PEFT replaces full fine-tuning with clever heuristics, allowing for training a tiny subset of model parameters while maintaining competitive performance with the standard models (Hu et al., 2021).

One of the most popular PEFT methods is Low-Rank Adaptation (LoRA)(Hu et al., 2021), which modifies only specific low-rank matrices within the model. By factorizing weight updates into smaller, manageable components, LoRA drastically reduces the number of trainable parameters. This method can be applied all layers, enabling efficient adaptation to new tasks without altering the model's core structure, all while achieving results comparable to full fine-tuning.

4 Method

This project is an *exploratory comparison study*, as it compares multiple pre-trained BERT-based models with each other and rates their performance on an F-1 score combined with a custom profit metric developed for this project. After conducting the literature review, selecting the appropriate model as a baseline was crucial for establishing a reference point against which other models could be compared.

4.1 BERT-based models

Bidirectional Encoder Representation for Transformers (BERT)(Devlin et al., 2019) is the most widely used encoder-based LLM with multiple sizes and variants. The basic variant comes with 12 attention heads, 12 layers, and an embedding size of 768 - totaling around 110M tunable parameters. It has been pre-trained on a large dataset consisting of two datasets:

- BookCorpus, a dataset containing books from various genres with around 800 million tokens, and
- The entirety of English Wikipedia, containing around 2.5 billion tokens.

This dataset has been used to learn two tasks:

1. Masked Language Modeling (MLM) - predicting randomly masked tokens in a sentence based on their surrounding context. This task is supposed to help the model understand the relationships between words and their context.
2. Next Sentence Prediction (NSP) - determining whether a given sentence logically follows another. This task is supposed to teach the model sentence-level understanding and relationships.

The recognition received not only in the domain of SA but in the whole field of NLP comes from its multiple advantages, namely, flexibility in size, superior performance, good explainability, simplicity and Variability

Due to its popularity, BERT has been chosen as the baseline model used for this project. All other models applied take advantage of a similar structure as this of the original BERT model with minor tweaks, such use of different data for pertaining or changed attention mechanisms. Since new methods were invented later on which improved the performance of BERT-based models significantly. Two main BERT variants used in the research are RoBERTa and DeBERTa, as seen in Table 1

Model	**Explanation**
RoBERTa	A robustly optimized version of BERT that removes the Next Sentence Prediction (NSP) task and trains on a larger dataset with longer training, resulting in better performance on various NLP tasks.
DeBERTa	Introduces disentangled attention, which separates content and position embeddings, and utilizes enhanced masking strategies. Often outperforms BERT and RoBERTa on complex tasks.

Table 1: Other Popular BERT Variants and Their Explanations

4.2 Domain-adaptive Pre-Training

While BERT and other foundational language models are pre-trained on general language corpora, research has shown that these models often struggle to understand domain-specific knowledge, as they lack the specialized vocabulary and context inherent to particular fields (Guo & Hauptmann, 2024). To address this limitation, several BERT-based models have been specifically pre-trained using domain-specific datasets, enabling them to better understand the terminology, nuances, and context of the target domain. For example, BioBERT (Lee et al., 2019) was pre-trained on large biomedical text corpora to enhance its ability to comprehend medical literature, while FinBERT (Araci, 2019) was pre-trained on the following data:

Dataset	**Description**
Financial PhraseBank[2]	A dataset of 5,000 financial sentences labeled with sentiment based on analyst agreement.
Thomson Reuters News [3]	A comprehensive financial news dataset containing articles and reports
Kaggle Stock Market Sentiment Dataset [4]	A collection of financial news headlines paired with stock price changes.

Table 2: Popular Financial Datasets for Training Sentiment Analysis Models

The use of domain-adaptive pre-training can significantly improve performance for tasks within the trained domain, as the model develops a deeper understanding of the domain's language patterns. In the context of this research, using domain-adapted models such as FinBERT allows for more accurate sentiment analysis of financial news, as the model is already familiar with the specific jargon and context used in financial reporting.

4.3 Training

The training process was implemented using two distinct approaches: traditional (full) fine-tuning and Parameter-Efficient Fine-Tuning (PEFT) using LoRA. For both approaches, we utilized the enhanced training techniques described in Section 2.3, including gradual unfreezing, slanted triangular learning rate(see Figure 2), and the ADAM optimizer.

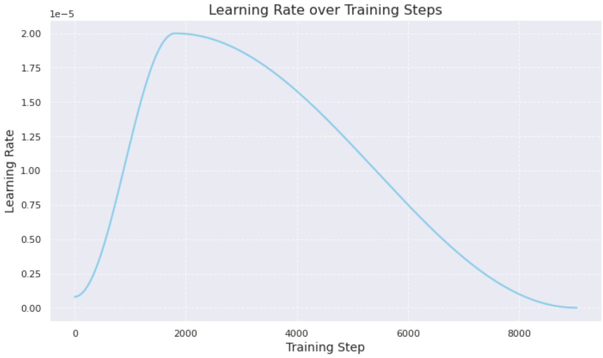


Figure 2: Cosine Slanted Triangular Learning Rate (SLTR) schedule used during model training. The learning rate initially increases during the warm-up period to reach an optimal learning zone, followed by a gradual decrease to fine-tune the model parameters.

For gradual unfreezing, we implemented a progressive layer unfreezing strategy based on the current training epoch:

$$n = \frac{currentEpoch}{allEpochs} * N$$

where n is the number of top layers to unfreeze and N is the total number of layers. This approach starts with all layers frozen except the classification head, then gradually unfreezes approximately one layer per epoch from top to bottom, ensuring the model retains its pre-trained knowledge while adapting to the new task.

The training was conducted on an A100 GPU from Habrok, with each model taking around 12 minutes to train, with around 2 hours needed to complete a standard hyperparameter grid search. To optimize for our imbalanced dataset, we employed weighted cross-entropy loss with class weights calculated based on the inverse frequency of each class in the training data. This issue is thoroughly described in the upcoming Data section.

Therefore, the loss function was defined as:

$$L = -\sum_{i=1}^{N} w_{y_i} \cdot \log(p_{y_i})$$

where w_{y_i} represents the class weight for the true label and p_{y_i} is the predicted probability for the correct class.

Model performance was evaluated using both standard metrics (ROC curves and F1 scores) and a custom profit metric designed to assess real-world trading performance. This dual evaluation approach ensured that the models were not only statistically sound but also practically viable for financial applications.

5 Experiments and Results

Building on the methodologies discussed in the previous sections, the focus is now shifted to the experiments conducted to evaluate the performance of the BERT-based models, including their domain-adapted variants.

5.1 Data

Due to the absence of a comprehensive dataset grounded in real-world trading environments, a custom dataset was created for this project. The dataset consists of 4,069 unique press releases from 118 biotech companies, collected from GlobeNewswire - the primary source for biotech company announcements. Biotech stocks were specifically chosen due to their high sensitivity to news, particularly around clinical trials, FDA approvals, earnings releases, and regulatory decisions, which can cause significant price movements within hours of announcements. Unlike other sectors where price changes may be gradual, biotech stocks often experience sharp movements based on binary outcomes (e.g., drug trial success/failure), making them ideal for sentiment analysis. The data collection pipeline involved three stages: First, company press releases were gathered through GlobeNewswire's RSS feeds, which provide real-time access to official company announcements. Second, corresponding stock price data was collected from Interactive Brokers, capturing price movements from one minute before to eight hours after each news release to measure the market's reaction. Finally, the data was reformatted and labeled on a five-point scale, with special attention paid to news categories that historically correlate with significant price movements, such as clinical study results (617 instances), earnings releases (416 instances), and regulatory announcements (279 instances). This comprehensive dataset provides a robust foundation for training models to predict market reactions to biotech news.

To ensure data quality and reduce noise, we implemented a rigorous topic pruning process. From the initial 50 news topics available in GlobeNewswire's categorization, only 33 were retained for the final dataset. Topics were filtered based on their historical correlation with significant price movements, requiring at least three instances of meaningful market impact (either positive or negative price movement, excluding neutral cases) to be included. This pruning helped focus the model on news categories that consistently influence stock prices, such as clinical trial results, regulatory decisions, and earnings releases, while eliminating noise from routine administrative announcements. The price movement thresholds were adapted from a paper by Aparicio et al., 2024. They scaled the labels according to market capitalization, acknowledging that smaller companies typically experience larger percentage moves: ±1% for companies with a market cap over \$10B, ±2% for \$2B-10B, ±3% for \$250M-2B, and ±4% for companies under \$250M market cap. This resulted in a three-class labeling scheme (positive, neutral, negative) with an uneven distribution: 64% neutral, and approximately 18% each for positive and negative labels (see Figure 3). To address this class imbalance during model training, we applied class weights of 1.9034, 0.5182, and 1.8356 for negative, neutral, and positive classes respectively, ensuring the model wouldn't simply learn to predict the majority class.

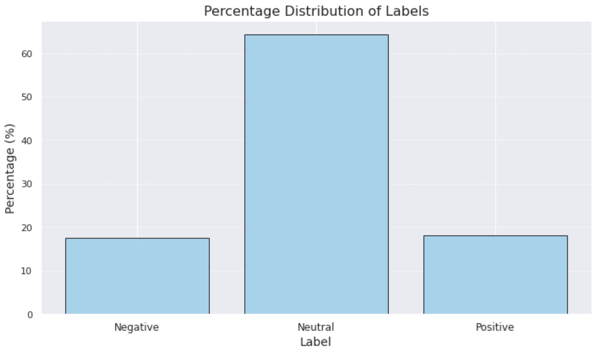


Figure 3: Distribution of sentiment labels in our dataset showing class imbalance. The neutral class dominates with 64% of samples, while positive and negative classes each represent approximately 18% of the data. This imbalance necessitated the use of weighted loss functions during training, with weights of 1.9034, 0.5182, and 1.8356 for negative, neutral, and positive classes respectively.

5.2 Evaluation

The performance of the models was assessed using multiple evaluation metrics. Initially, the F1 score(harmonic mean of precision and recall) is calculated to quantify the models' overall classification ability. The F1 score is particularly valuable as it provides a balanced measure of a model's ability to correctly identify both positive and negative instances.

In addition to the F1 score, ROC curves were also used to further evaluate model performance. ROC curves are particularly useful as they visualize the trade-off between the true positive rate and false positive rate across different threshold values, helping to assess the model's ability to distinguish between class distributions. This metric combined with the F1 score gives a comprehensive summary of model performance from a classical point of view.

Furthermore, to evaluate the real-world applicability of our models, we developed a custom profit metric that simulates actual trading decisions based on model predictions. Unlike traditional metrics like F1-score or accuracy that treat all misclassifications equally, this metric accounts for the magnitude of price movements and the directional correctness of predictions, making it more relevant for practical trading applications.

The profit metric works by simulating a simple trading strategy:

- For negative predictions (label 0): Take a short position, profiting from price decreases
- For neutral predictions (label 1): No position taken
- For positive predictions (label 2): Take a long position, profiting from price increases

The profit for each prediction is calculated as:

$$profit_i = \begin{cases} -\frac{close_i - buy\_in_i}{buy\_in_i} & \text{if prediction} = 0 \text{ (short)} \\ 0 & \text{if prediction} = 1 \text{ (neutral)} \\ \frac{close_i - buy\_in_i}{buy\_in_i} & \text{if prediction} = 2 \text{ (long)} \end{cases}$$

where buy_in_i is the first available stock price after news release and $close_i$ is the price after 8 hours. The total profit is then calculated as the sum of all individual trade profits:

$$total\_profit = \sum_{i=1}^{N} profit_i$$

This metric provides a more practical assessment of model performance than traditional classification metrics. For example, a model might achieve high accuracy by correctly predicting many small price movements while missing the few large movements that matter most for trading profitability. Our profit metric, in contrast, directly measures the model's ability to capture profitable trading opportunities, regardless of their frequency.

Importantly, this metric was not used during model training but rather as a post-training evaluation tool to assess real-world applicability. This separation ensures that the models learn general patterns in the data (optimizing for F1-score) while still being evaluated on their practical trading performance.

5.3 Experimental Details

The models used in this research have been obtained from the HuggingFace Python Library(Wolf et al., 2020). Furthermore, to ensure good coding practices, the custom HF trainer used for fine-tuning has minimal modifications that were necessary to incorporate the training methods as explained in the previous sections. Other main libraries used in this research were PyTorch, NumPy, Pandas, and Seaborne. To ensure reproducibility, the random seeds were set for all of the libraries used in the research.

The train-test split used in this research was set to 90/10, meaning that the test set contained 402 data points while the model was trained on 3618 examples. The split was done with stratification, ensuring an even label distribution for both of the datasets.

The hyperparameters were carefully tuned through grid search, with the final configuration including:

Hyperparameter	**Value**
Learning rate schedule	2e-5 → 1e-4 → 0
Epochs	40 with early stopping
Weight decay	0.01
Batch size	16
Gradient accumulation steps	2

Table 3: Fine-tuning hyperparameter configuration

For the LoRA implementation, we used an inner dimension r as a hyperparameter that affects the number of parameters in the low-rank matrices. This approach significantly reduced the number of trainable parameters while maintaining model performance. The LoRA configuration included:

Hyperparameter	**Value**
Rank (r)	32
LoRA alpha	32
LoRA dropout	0.1

Table 4: LoRA hyperparameter configuration

5.4 Result

After the fine-tuning process was concluded we obtained following results.

Model	F1-score	Profit
BERT (B)	0.052	-94%
BERT	0.56	67%
BERT (LoRA)	0.55	12%
DeBERTa	0.565	15%
FinBERT	0.576	**168%**
DeBERTa (F)	**0.646**	34%
BioBERT	0.595	41%
RoBERTa	0.603	16%

Table 5: Performance of different models on the evaluation dataset (F - fine-tuned on financial data before training, B - not fine-tuned, baseline model).

5.5 Analysis

Our experiments yielded several interesting findings across different model architectures and training approaches. The results demonstrate the need for traditional metrics like the F1 score and real-world performance metrics like profit. FinBERT emerged as the overall winner with an F1 score of 0.576 and an profit metric of 168% (even with different hyperparameter settings, FinBERT was consistently achieving a profit of > 150%). This performance significantly outpaced other models, including those with higher F1 scores, highlighting the impact of domain-specific pre-training for financial applications. Interestingly, DeBERTa fine-tuned on the Financial PhraseBank dataset (Malo et al., 2013) achieved the highest F1 score (0.646) but only generated a 34% profit.

5.5.1 Domain Adaptation Analysis

The results demonstrate a clear pattern regarding domain adaptation:

- **Financial Domain Adaptation**: Models with financial domain exposure (FinBERT, DeBERTa FT Fin News) consistently outperformed the baseline BERT model (F1: 0.56) in classification accuracy.
- **Biomedical Domain Adaptation**: BioBERT achieved a strong F1 score of 0.595 and a profit of 41%, showing that understanding biomedical terminology contributes positively to performance, though not as significantly as financial domain knowledge.

5.5.2 ROC Curve Analysis

The ROC curves reveal important insights about model behavior across different classes:

- **BioBERT** showed balanced performance across classes (Area under the Curves (AUC): 0.60, 0.62, 0.58), indicating consistent ability to handle biomedical terminology.
- **DeBERTa fine-tuned** demonstrated stronger performance on all cases (Class 0 AUC: 0.63; Class 1 AUC: 0.65; Class 2 AUC: 0.64), suggesting better calibration for identifying non-market-moving and market-moving news.
- **Baseline BERT** showed relatively uniform but weaker performance (AUC around 0.51-0.53), confirming the benefits of domain-specific training.

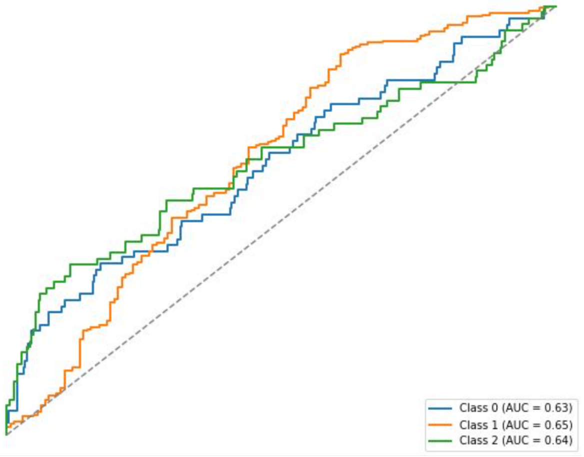


Figure 4: Receiver Operating Characteristic (ROC) curves for DeBERTa fine-tuned model across three sentiment classes. The curves demonstrate the model's discrimination ability, with Class 1 (neutral) showing the strongest performance (AUC: 0.65), followed by Class 0 (negative) and Class 2 (positive). The diagonal dashed line represents random classifier performance (AUC: 0.5). The higher AUC values across all classes indicate the model's ability to effectively distinguish between different sentiment categories, particularly for neutral sentiment identification.

5.5.3 Efficiency vs Performance

The BERT LoRA implementation achieved an F1 score of 0.55 and a profit of 12%, demonstrating that parameter-efficient fine-tuning can maintain reasonable performance while significantly reducing computational requirements. While this represents a performance trade-off compared to full fine-tuning (BERT: 0.56 F1, 67% profit), the minimal degradation in performance suggests LoRA as a viable option for resource-constrained environments.

6 Conclusion

6.1 Key Findings

1. Domain-specific pre-training (FinBERT) proved more valuable than superior architectures (DeBERTa, RoBERTa) for financial applications.

2. Traditional metrics (F1 score) don't necessarily correlate with real-world trading performance, emphasizing the importance of domain-specific evaluation metrics.

3. The baseline BERT model's strong profit performance (67%) suggests that basic language understanding combined with fine-tuning can capture significant market signals.

4. Parameter-efficient methods like LoRA offer a viable alternative with acceptable performance trade-offs.

6.2 Limitations

The initial project setup used a five-label classification scheme (very positive, positive, neutral, negative, very negative). However, the model struggled to recognize the inherent relationships between these classes (e.g., "very positive" as a specific case of "positive"), which adversely affected its performance. As a result, the team switched to a three-label scheme, which, while simpler, contains less detailed information and makes it more challenging for the model to optimize for profit. This limitation is discussed further in Section 6.3.1.

This issue underscores a more significant challenge: data noise. Since the entire dataset was collected, preprocessed, and labeled by the research team, the quality of the data was affected by time constraints and subjective decisions. One of the main issues noticed during the testing of the algorithms was that if a news article was highly positive to a company yet published outside of market working hours, the price would be lifted up by the brokers so that the opening price would sometimes partially reflect those news before trading was made available. This happens mostly due to the way those stock markets operate, but our model gets punished for making the correct prediction ("buy"). The project reveals several challenges, suggesting that any successful application would require either a larger, more comprehensive dataset or more refined preprocessing techniques.

6.3 Future Research

6.3.1 Regression

During our research, we observed that the models showed no ability to detect similarities between related labels (i.e., positive and highly positive when trying to train with 5 label), leading to training inefficiency. Converting the problem from classification to regression could address this limitation by allowing the model to learn continuous relationships between news sentiment and price movements to better capturing the magnitude of market reactions.

6.3.2 Mixture of Experts / Boosting

Our results with FinBERT and BioBERT suggest that domain expertise matters significantly. Future work could explore ensemble methods where:

- Multiple pre-trained models (FinBERT and BioBERT) work together on the same data point to reach a consensus looking at the problem from multiple angles.
- A summarizer model could be implemented to amplify the tone of press releases
- Boosting techniques could be applied to add a feature extraction layer on before of the classification.

6.3.3 Decoder-based Architectures

Given the rise of decoder-based architectures and their increasing prominence in NLP research, future work should investigate the comparison of encoder-only vs decoder-based approaches for financial text, the potential for zero-shot and few-shot learning capabilities, and exploration of hybrid architectures combining encoder and decoder strengths

6.4 Final words

To summarize, this research highlights the potential of combining domain-specific pre-training with innovative fine-tuning methods to achieve models that can reliably turn a profit in the market while outpacing potential competitors. By outlining limitations and providing future research directions, this work sets a strong foundation for advancing automated trading systems and, more generally, delivering impactful solutions in the financial domain.

Acknowledgments

Multiple major design decisions made in this research such as applying the custom profit metric or

framing the problem as sentiment analysis were suggested by a former investment fund manager with 20+ years of experience in finance, not affiliated with the University of Groningen. The expert's identity was requested to be kept anonymous.

References

Aparicio, V., Gordon, D., Huayamares, S. G., & Luo, Y. (2024). Biofinbert: Finetuning large language models (llms) to analyze sentiment of press releases and financial text around inflection points of biotech stocks. https://api.semanticscholar.org/CorpusID:267068871

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *ArXiv*, *abs/1908.10063*. https://api.semanticscholar.org/CorpusID:201646244

Chen, S., Hou, Y., Cui, Y., Che, W., Liu, T., & Yu, X. (2020). Recall and learn: Fine-tuning deep pretrained language models with less forgetting. https://arxiv.org/abs/2004.12651

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805

Ding, H., Li, Y., Wang, J., & Chen, H. (2024). Large language model agent in financial trading: A survey. https://arxiv.org/abs/2408.06361

Du, K., Xing, F., Mao, R., & Cambria, E. (2024). Financial sentiment analysis: Techniques and applications. *ACM Comput. Surv.*, *56*(9). https://doi.org/10.1145/3649451

Guo, T., & Hauptmann, E. (2024). Fine-tuning large language models for stock return prediction using newsflow. *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:271432060

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:40100965

Hu, J. E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *ArXiv*, *abs/2106.09685*. https://api.semanticscholar.org/CorpusID:235458009

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). Biobert: A pretrained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*, 1234–1240. https://api.semanticscholar.org/CorpusID:59291975

Malo, P., Sinha, A., Takala, P., Korhonen, P., & Wallenius, J. (2013, July). *Financialphrasebank-v1.0*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need. https://arxiv.org/abs/1706.03762

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., . . . Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing. https://arxiv.org/abs/1910.03771