A Multimodal Approach to Stock Directionality: Transformers, Regression, and Social Media

Adewale-Young Adenle
Ohio State University
adenle.4@osu.edu

Pranav Khetarpal
Ohio State University
khetarpal.12@osu.edu

Adithya Chandrashekar
Ohio State University
chandrashekar.26@osu.edu

Abstract

This study explores the integration of sentiment analysis from financial tweets with time-series forecasting to predict directional stock movement. We fine-tune two transformer models, general-purpose BERT and domain-specific FinancialBERT, on financial sentiment datasets (StockEmotions and Zeroshot), and evaluate their performance both independently and when paired with an AutoRegressive (AR) model. Despite FinancialBERT's domain specialization, BERT outperforms it in sentiment classification, likely due to its pretraining on more diverse, conversational text better aligned with the informal nature of tweets. While the AR model performs well on its own, both hybrid models (BERT+AR and FinancialBERT+AR) achieve identical directional accuracy of 88.57%, surpassing all standalone models. This suggests that confidence-based integration can selectively enhance time-series predictions using sentiment signals, while mitigating differences between sentiment models in hybrid setups.

1 Introduction

Financial markets are influenced by a variety of factors including the economy, earnings reports, geopolitical events, and investor sentiment. Traditional stock prediction models typically rely strictly on numerical data and often overlook the impact of financial news, analyst reports, and social media discussions. The rise of digital communication has created vast amounts of unstructured financial text that could significantly affect investment decisions, highlighting the need to integrate qualitative data into predictive models (Pagolu et al., 2016).

Advancements in natural language processing (NLP) and deep learning, particularly transformer-based models such as BERT (Devlin et al., 2019), FinancialBERT (Hazourli, 2022), and FinBERT, have improved sentiment analysis and the contextual understanding of financial text. Previous research has shown a strong correlation between public sentiment and stock price fluctuations (Pagolu et al., 2016; Palomo, 2023), yet challenges remain due to domain-specific vocabulary, less explicit sentiment, and the inherent volatility of financial markets.

This study addresses the following research questions:

1. Which performs better on sentiment analysis of stock-related tweets: a fine-tuned BERT model or a fine-tuned FinancialBERT model?

2. How does an AutoRegressive (AR) model perform in predicting whether a stock's price goes up or down (directional accuracy) based on closing prices?

3. How does the directional accuracy of stock movement predictions change when combining outputs from a BERT or FinancialBERT model with an AR model, compared to using the AR model alone?

By leveraging sentiment classification, contextual embeddings, and event-driven text modeling, this research evaluates the effectiveness of fine-tuning different transformer models for financial forecasting, as well as their performance when each is paired with an AutoRegressive model, which employs classical statistical forecasting techniques (Lee et al., 2023).

2 Motivation/Objective

The rapid increase in financial news, social media discussions, and analyst reports has created a vast repository of unstructured textual data that can offer valuable signals for stock price prediction. While conventional models rely on historical price trends and financial indicators, they often fail to capture market reactions driven by sentiment and breaking news.

This study aims to enhance stock price forecasting by developing and comparing predictive frameworks that integrate financial text analysis with time-series modeling. Specifically, we aim to:

1. Develop a standalone AutoRegressive (AR) model and combine it independently with both the fine-tuned BERT and fine-tuned FinancialBERT models.

2. Fine-tune FinancialBERT on financial tweet data for sentiment classification and integrate it with the AR model, forming the (FinancialBERT + AR) hybrid.

3. Fine-tune BERT on financial tweet data and combine it with the AR model, forming the (BERT + AR) hybrid model.

4. Compare the performance of both hybrid models to evaluate the effectiveness of fine-tuning domain-specific versus general-purpose language models for financial sentiment classification in stock prediction.

By integrating sentiment analysis, contextual embeddings, and event-driven text classification, this study contributes to the fields of financial NLP and algorithmic trading. The findings will provide insights into how different transformer-based models perform when fine-tuned on financial text and combined with classical time-series modeling, offering investors improved data-driven decision-making tools.

3 Related Work

One of the most relevant recent studies in this domain is *Tweet Sentiment Analysis to Predict Stock Market* by Christian Palomo of Stanford University (Palomo, 2023). This research developed an NLP model to predict stock market movement through the analysis of Twitter tweets, employing a transformer-based neural network for sentiment analysis.

Stock market movement prediction in this study was defined as forecasting whether a stock's price would move up or down from the previous day's closing price to the current day's closing price. A prediction $Y_t = 1$ for a given stock s indicated that the closing price of s was greater today than the previous day, while a prediction of 0 indicated that the closing price was less than or equal to the previous day's closing price. To make predictions, the model analyzed both the historical stock information from the previous T trading days and the tweets pertaining to the stock over that same period.

The tweets were classified according to sentiment: Bullish, Bearish, or Neutral.

- **Bullish** classification refers to buying a stock in anticipation of a price rise.

- **Bearish** refers to selling a stock expecting a price drop.

- **Neutral** reflects no particular opinion on buying or selling.

The model was trained on the "zero-shot/twitter-financial-news-sentiment" dataset from Hugging Face, which we also used (zeroshot, 2024). It is a fine-tuned FinancialBERT-SentimentAnalysis model, building upon and refining the FinancialBERT model of Hazourli (Hazourli, 2022).

Performance was evaluated using accuracy and F1 score (based on precision and recall) against baselines, including:

- FinancialBERT-Sentiment-Analysis

- FinancialBERT-Sentiment-Analysis without tweet preprocessing

- Twitter-roBERTa-base for Sentiment Analysis

- A standard BERT binary classifier

The model achieved an accuracy of 0.843 and an F1 score of 0.843 when evaluated on stock data and tweets for Tesla (TSLA) from August 2022 to December 2022.

Another relevant study was conducted by Pagolu et al. (Pagolu et al., 2016), which explored sentiment classification of stock-related tweets using Random Forest, Logistic Regression, and SMO models. This study was significant because, at the time, NLP-based approaches for stock market prediction were still emerging, with most prior methods relying heavily on historical stock price data. Pagolu et al. diverged from this trend by leveraging a relatively small but high-quality, human-labeled dataset of 3,216 tweets to train their models.

Their best-performing model, Random Forest, achieved the following results using two different feature extraction techniques:

- **Word2Vec Representation:** Accuracy: 70.18%, Precision: 0.711, Recall: 0.702, F1-score: 0.690
- **N-gram Representation:** Accuracy: 70.49%, Precision: 0.719, Recall: 0.705, F1-score: 0.694

This study relates to our work as it demonstrates the feasibility of performing sentiment analysis on stock-related tweets using methods of data extraction and preprocessing similar to ours.

While both (Palomo, 2023) and (Pagolu et al., 2016) explored not only sentiment analysis, but also sentiment-based stock movement prediction through combined model method approaches, their implementations primarily relied on classification techniques, such as Random Forest, to predict directional movement, rather than classical statistical forecasting techniques. In contrast, our approach explicitly integrates fine-tuned sentiment models (BERT, FinancialBERT) with AutoRegression (AR) to directly forecast price trajectories using time-series dynamics.

By integrating sentiment classification with classical time-series forecasting techniques such as AutoRegression, we aim to investigate whether combining deep learning-based sentiment signals with statistical models can lead to improved predictive performance compared to using either approach independently. This novel aspect of our study differentiates our work from previous research and could provide valuable insights into the effectiveness of hybrid approaches for financial market forecasting.

4 Methodology

4.1 Dataset

Our financial text analysis pipeline consists of several key components, each contributing to the preprocessing and integration of financial data. The main components are Yahoo Finance data, preprocessing with StockEmotions, Zeroshot, and processing using GPT-4o.

4.1.1 Preprocessing with StockEmotions

We processed the StockEmotions dataset, which was originally obtained from the StockEmotions GitHub page (Lee et al., 2023).

Preprocessing Steps

- **Cleaning:** Emojis and the ticker symbols, which appear at the beginning of tweets and are prefixed with a dollar sign ($), were removed, along with other special characters, but mentions of stock tickers and dollar amounts in the tweet body were preserved. White space was also normalized across each tweet.
- **Sentiment Labels:** The sentiment of each tweet was classified as either "bullish" or "bearish", corresponding to positive and negative sentiments, respectively, and these labels were retained in the processed data.

This preprocessing ensures that only realistic tweet content is provided to the model during training and inference.

4.1.2 Preprocessing with Zeroshot

We used the publicly available Zeroshot stock sentiment dataset through HuggingFace (zeroshot, 2024). This dataset required preprocessing to prepare it for training and evaluation.

Preprocessing Steps

- **Ticker Extraction and Normalization:** Stock tickers, which appear at the beginning of tweets and are prefixed with a dollar sign ($), were extracted and cleaned. Tweets with multiple tickers were duplicated to ensure each ticker had its own entry.
- **Text Cleaning:** We removed hyperlinks, special characters, and normalized whitespace in the tweet text. If no stock ticker was present at the start of a tweet, we assigned a placeholder ticker "UNKNOWN".
- **Sentiment Labels:** The sentiment of each tweet was classified numerically as either "bullish", "bearish", or "neutral". Tweets labeled as neutral were strictly excluded to maintain a clear binary classification boundary, and to align with data that contained only two labels (without neutral), to enforce consistency and accuracy.

This process made sure ticker associations were clearer as well as cleaner input text for model fine-tuning, while preserving the original label annotations.

4.1.3 Preprocessing with GPT-4o

The GPT-4o preprocessor is responsible for extracting ticker symbols from financial text and classifying sentiment for those tweets from the Zeroshot

dataset for which the relevant stock was unknown and only when the preprocessor function using GPT-4o was able to confidently identify the relevant stock. The preprocessing pipeline handles CSV files containing financial texts, ensuring ticker symbols are appropriately assigned and numerical sentiment labels are converted to text format.

Features of Preprocessing:

- **Ticker Symbol Extraction:** Uses GPT-4o to identify stock ticker symbols in financial text.
- **Batch Processing:** Efficiently processes texts in batches of 20 to optimize API usage.
- **Sentiment Classification:** Converts numerical sentiment labels (0 $\rightarrow$ "bearish", 1 $\rightarrow$ "bullish") into text format.
- **Filtering:** Ignores entries with neutral sentiment (label 2) from the dataset.
- **Structured Output:** Returns the results in a standardized CSV format with columns for the ticker symbol, sentiment label, and original text content.

4.1.4 Combining StockEmotions and Zeroshot

After independently processing the Zeroshot and StockEmotions datasets, we merged them to form our final dataset. The training dataset from Zeroshot was combined with the training dataset from StockEmotions, and the same was done for the validation data. The test data was solely used from StockEmotions, however.

Dataset Composition

- The training dataset consists of 11,496 examples.
- The validation dataset consists of 1,852 examples.
- The test dataset consists of 1,000 examples.

This final dataset, which includes ticker symbols, cleaned tweets, and sentiment labels, was used for fine-tuning the BERT-type models and evaluating their performance.

4.1.5 Yahooquery Data

Yahooquery serves as a utility module to interface with Yahoo Finance and retrieve historical financial market data (Aroussi, 2017). It allows us to fetch, process, and store this data in a standardized CSV format for further analysis.

For this project, we used Yahooquery to construct a dataset of 34 stocks, containing their daily closing prices from January 2010 to December 2020. The dataset is arranged such that each row represents a trading day, with the date listed in the first column and each subsequent column containing the closing price of one of the 34 stocks. This structure ensures chronological alignment and consistency across all stocks, and the resulting data resembles a CSV-formatted table as shown below.

Date	**AAPL_ CLOSE**	**AMT_ CLOSE**	**..._ CLOSE**	**XOM_ CLOSE**	**^GSPC_ CLOSE**
01/04/2010	Price	Price	···	Price	Price

Table 1: Example format of the stock price dataset

4.2 Training

4.2.1 Training BERT and FinancialBERT

This component of the pipeline focuses on fine-tuning transformer-based models for sentiment classification of financial tweets. We experiment with two variants: the general-purpose BERT model (Devlin et al., 2019), and FinancialBERT, a domain-adapted variant that is pre-trained on a large-scale financial corpora (Hazourli, 2022).

BERT uses a bidirectional transformer architecture, which allows it to capture context from both preceding and following tokens in a sequence. FinancialBERT extends this capability by incorporating domain-specific knowledge through pre-training on Reuters financial news, Bloomberg articles, SEC filings, and earnings call transcripts. This makes FinancialBERT particularly well-suited for capturing financial jargon.

Even though FinancialBERT is originally developed and configured for masked language modeling, we adapt both BERT and FinancialBERT for binary sentiment classification using the BertForSequenceClassification architecture with two output labels: bullish and bearish.

Fine-Tuning Setup We use the AdamW optimizer with weight decay and optional cosine or linear learning rate scheduling. Training is specifically performed for a configurable number of

epochs, and early stopping is utilized based on validation F1 score to reduce the chances, as well as effects of, overfitting. The checkpoint with the best validation performance is saved and is used for testing. The optimal training configuration for BERT and FinancialBERT is listed in Table 2.

Configuration	BERT	FinancialBERT
Loss function	CrossEntropyLoss	CrossEntropyLoss
Batch size	8	16
Learning rate	2e-5	1.5e-5
Scheduler	linear	cosine
Weight decay	0.01	0.01
Optimizer	AdamW	AdamW
Max epochs	2	5
Patience epochs	2	2

Table 2: Optimal training configuration for BERT and FinancialBERT.

During each epoch, we compute both training loss and validation metrics (loss and weighted F1). The final model is evaluated on a held-out test set for generalization performance.

Model checkpoints are stored separately for BERT and FinancialBERT. These models are later integrated with the downstream time-series forecasting module.

4.2.2 Training the AutoRegressive Model

This component of the pipeline focuses on training an AutoRegressive (AR) model for univariate time-series forecasting, using smoothed and normalized closing prices for 34 stocks. We use AR models to capture autoregressive dependencies within each individual stock's price history and evaluate their predictive power both in terms of regression error and directional accuracy (Dalal et al., 2019).

Feature Engineering The only feature used for each stock is its historical daily closing price, forming a univariate time series. These values are:

- Aligned by trading date across all stocks
- Normalized using z-score standardization (using the training set's mean and standard deviation)
- Smoothed using simple moving average (SMA) with a rolling window of 19 days (selected through hyperparameter search)

This preprocessing ensures stability in the input signal and mitigates daily volatility before model training.

Dataset Splits The dataset spans over a decade and is split chronologically to avoid temporal leakage:

- Training set: January 2010 – December 2018
- Validation set: January 2019 – December 2019
- Test set: January 2020 – December 2020

The 2020 test period aligns with the tweet dataset used for sentiment modeling, enabling consistent cross-modal evaluation in the combined model approach.

Lag Selection and Hyperparameter Optimization The AR model includes a lags parameter indicating how many prior time steps are used to predict the next day. We performed a grid search over a range of lag values (8 to 19) and window sizes (3 to 15), evaluating average directional accuracy on the validation set. The best-performing configuration found was:

- Window size: 19 (for smoothing)
- Lag size: 8 (for AR modeling)

This lag length captures meaningful short-term memory without irregularities.

Model Training and Evaluation Each AR model is trained independently for every stock using statsmodels.tsa.ar_model.AutoReg (Seabold and Perktold, 2010). The training procedure includes:

- Iterative forecasting on validation and test sets using a rolling window update strategy
- Prediction of daily closing price for the next day

For each test example, the model is retrained using all available historical data up to that point, simulating a real-world rolling forecast scenario.

We evaluate two metrics:

- **Mean Squared Error (MSE)**: Measures regression accuracy of price predictions
- **Directional Accuracy (DA)**: Measures how often the model correctly predicts whether the price will rise or fall relative to the previous day

Both metrics are averaged across all 34 stocks.

Test Set Evaluation and Output After concatenating the training and validation sets, we retrain the AR models and generate predictions on the 2020 test set. Directional predictions and actual movements are saved in CSV format. These are later used for combining statistical and sentiment-based signals for hybrid prediction.

Previous Approach The decision to proceed with an AutoRegressive (AR) model was made after preliminary experimentation with the Vector AutoRegression (VAR) model. While VAR is designed to capture dependencies across multiple time series (Stock and Watson, 2001), making it useful for modeling inter-stock relationships, we found that this approach was not well suited to our task. Specifically, our goal was to predict the directional movement of individual stocks rather than model relationships between stocks. In this context, VAR's attempt to learn cross-stock dependencies often introduced noise rather than meaningful signal, ultimately decreasing performance. Moreover, our initial tests with VAR produced relatively poor results, further confirming that its structure did not align with the needs of our experiment. The AR model, however, allowed us to isolate and model each stock's historical trajectory independently. This made it a more focused and interpretable approach, making sure that predictions were based solely on the past behavior of the target stock rather than influenced by unrelated or weakly correlated areas.

4.3 Inference

4.3.1 BERT and FinancialBERT

To evaluate the baseline performance of our models before fine-tuning, we first tested the existing FinancialBERT and BERT models on our test dataset. We downloaded both models and utilized the BertForSequenceClassification configuration, which adds a classification head to the base BERT transformer, to perform sentiment analysis on the tweets in the test set. The baseline results are as follows:

- **FinancialBERT:** F1-score: 0.505, Test accuracy: 0.506
- **BERT:** F1-score: 0.134, Test accuracy: 0.444

From these initial evaluations, FinancialBERT outperformed BERT in classification accuracy, suggesting that it may be more suited for financial sentiment analysis. After fine-tuning the BERT and FinancialBERT models using the configurations detailed in Section 4.2.1, we evaluated their performance on a test dataset composed of tweets from the year 2020. The models achieved the following results:

- **Fine-tuned BERT model:** F1-Score: 0.7602, Accuracy: 0.7600
- **Fine-tuned FinancialBERT model:** F1-Score: 0.7055, Accuracy: 0.7060

Interestingly, the fine-tuned BERT model outperformed the financial domain specific FinancialBERT model. While this may seem counterintuitive considering FinancialBERT's training on financial texts, this result is a product of the differences in the pre-training corpora and the style of text found in the tweets.

BERT was pre-trained on a large and diverse corpus of text from many different or general domains, such as Wikipedia, which includes informal and conversational language similar to that used in tweets. This pre-training helped the model adapt more effectively during fine-tuning to the unstructured or informal language used in tweets about stocks. In contrast, FinancialBERT was pre-trained on formal financial documents, which tend to follow structured and technical language patterns. As a result, FinancialBERT struggled to adapt to the more casual and varied nature of stock-related tweets.

It is also important to note the difficulty of the test dataset itself. Many of the tweets contained ambiguous or misleading language, and lacked clear sentiment hints. Below are a few examples where each model misclassified the tweet sentiment:

BERT failure examples:

- "with all time highs again the market seems to enjoy war very much"
  - true label: bearish
  - predicted label: bullish
- "omg this can't get past 230"
  - true label: bullish
  - predicted label: bearish

FinancialBERT failure examples:

- "Imagine Elon says something magical?"

  - true label: bullish
  - predicted label: bearish

- "anyone believes it'll hit $496 today??"
  - true label: bearish
  - predicted label: bullish

These examples highlight the inherent challenge in sentiment analysis of stock-related tweets: the language is often dependent on context, indirect, or mock. Without deeper contextual understanding or user-specific data, the models have difficulty accurately interpreting the true intent or tone of the tweet.

Given these challenges, we do not expect either model to achieve perfect accuracy on this task. Nevertheless, the results suggest that a more general-purpose language model like BERT may be better suited to capturing the subtleties of tweet language than a domain-specific model like FinancialBERT, at least in its current form and without additional adaptations.

4.3.2 AutoRegressive Model Inference

To evaluate the performance of our times series model in isolation, we ran inference using a trained AutoRegressive (AR) model on a held-out test set. We focus primarily on directional accuracy as our evaluation metric, since the objective of this study is to predict whether a stock's price will rise or fall, not to estimate the precise price. This is consistent with our sentiment classification setup, where tweets are labeled as bullish or bearish, categories that correspond to actionable trading signals. As a result, metrics like Mean Squared Error (MSE) are calculated but omitted, as they do not align with the goal of our model and experiment.

Evaluation Setup Following training (Section 4.2.2), we merged the training and validation sets to form a combined historical sequence for each stock. The AR model is retrained on this extended sequence to maximize its available context before predicting on the 2020 test period.

Inference was performed by using a rolling forecast strategy. At each time step t, we:

1. Fit a new AR model using all available data up to time t
2. Generated a 1-day-ahead prediction $\hat{y}_{t+1}$ using the model
3. Compared this prediction to the actual $\hat{y}_{t+1}$ in order to calculate the Mean Squared Error (MSE) and Directional Accuracy.

This process simulated realistic forecasting conditions where only the past data was accessible.

Output Metrics For each of the 34 stocks, we computed the following:

1. **Mean Squared Error (MSE)**:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

   where y_i is the actual value and $\hat{y}_i$ is the AR model prediction.

2. **Directional Accuracy (DA)**: The percentage of times the model correctly predicts the direction (up/down) of the stock's movement:

$$\text{DA} = \frac{1}{n}\sum_{i=1}^{n} 1[\text{sign}(\hat{y}_i - y_{i-1}) = \text{sign}(y_i - y_{i-1})],$$

   where y_i is the actual value, y_{i-1} is the actual value for yesterday, $\hat{y}_i$ is the AR model prediction, and 1 is for the indicator function.

The predicted and actual movement directions are then stored in CSV format for future analysis/reference.

Model Configuration We use the previously selected optimal configuration from validation:

1. Lag: 8 days
2. Smoothing window: 19 days (simple moving average)

Results The directional accuracy values for all 34 stocks assessed are included in Appendix A below.

Aggregate values are averaged across all 34 stocks to assess the overall performance of the model, with the Average Directional Accuracy being 0.8773466833541925. These directional movement outputs from evaluation are used for comparison with BERT-based sentiment predictions and their hybrid integration.

4.3.3 Combined Approach

The combined approach integrates two distinct methodologies to achieve more accurate financial predictions. This hybrid model leverages both natural language processing capabilities and time-series forecasting techniques to capture different aspects of market behavior.

The transformer components form the first pillar of this approach, utilizing BERT and FinancialBERT transformer architectures specifically fine-tuned for sentiment analysis on stock-related tweets. The second pillar consists of an AutoRegressive component that implements time series forecasting using the AutoRegressive model.

The algorithm used in this approach takes two inputs for each stock on each day: the average sentiment score from either the BERT or FinancialBERT model, and the predicted stock movement direction from the AutoRegressive (AR) model.

The integration module combines predictions from both components using a confidence-based decision mechanism. When the sentiment model and the auto-regressive model agree, the system increases its confidence in the shared prediction. When the models disagree, the confidence is reduced based on the intensity of the sentiment signal. This adaptive thresholding strategy enables the model to make more informed predictions by leveraging the strengths of both price-based trends and textual sentiments. As a result, the final directional prediction reflects both technical market patterns and investor sentiment, leading to improved accuracy over either method alone.

To see a detailed explanation of the algorithm used in integration, see Appendix D, which also includes pseudocode.

The data used to evaluate the combined model consisted of the output predictions from the BERT, FinancialBERT, and AutoRegressive (AR) models on their respective test datasets. Specifically, the sentiment models (BERT and FinancialBERT) produced an average sentiment score for each tweet, aggregated by stock and by day, using tweets from the year 2020. In parallel, the AR model generated predicted stock movement directions for each day in 2020 across all the stocks represented in the test dataset. These sentiment scores and directional forecasts formed the inputs to the combined model during evaluation.

The combined approach had the following results on its evaluation set, which consisted of the average sentiment score for 350 stock and date combinations, and the corresponding 350 predicted directional movements from the AutoRegressive model.

To evaluate the performance of our hybrid model, we tested both BERT + AutoRegression (AR) and FinancialBERT + AR approaches on the same test set. Both combinations achieved an identical directional accuracy of 0.8857, outperforming each individual component used in isolation. AR alone yielded 0.8773 accuracy, while BERT achieved 0.7600 and FinancialBERT achieved 0.7060. The performance for the hybrid models was achieved using a Base Confidence of 0.65, a Confidence Threshold of 0.10, and a Boost Confidence of 0.10, for both combinations. This improvement confirms that incorporating sentiment predictions can meaningfully improve traditional time-series forecasting.

While BERT showed stronger standalone performance than FinancialBERT, their contributions to the hybrid model were effectively equalized by the model's confidence-aware fusion strategy. In our implementation, the AR model always makes the initial prediction, and sentiment is used to adjust confidence in that prediction. If the adjusted confidence falls below a set threshold, the model flips the AR prediction. This means that sentiment doesn't necessarily act as an override, but rather more as a veto mechanism when AR predictions appear uncertain in light of strong opposing sentiment.

Because both BERT and FinancialBERT tended to produce sentiment scores on a similar subset of dates, their net effect on the final predictions was nearly identical, resulting in matching hybrid accuracy scores. This highlights a key insight: even if sentiment models differ in raw accuracy, their confidence patterns can lead to similar influence in hybrid systems when used selectively.

Overall, this result shows the value of confidence-based decision mechanisms: by starting from a stable AR prediction and only adjusting when sentiment is compelling, the model benefits from text signals when they matter most.

Results Directional accuracy for each of the stocks is listed in Appendices B and C.

Acknowledgments

ChatGPT was used in preparing this report. It was specifically used for wording purposes to introduce

more clarity, formalize the tone and language as appropriate for an academic paper, and finding specific phrases to convey the message that is trying to be explained in certain parts of the report.

Contributions

All team members contributed equally to the project, with each member taking the lead on different components. Research on BERT and FinancialBERT models was primarily conducted by Pranav and Adewale, while Adithya focused on researching AutoRegressive models. All three team members collaborated on researching related prior work. Data preparation for historical stock price data was led by Adewale. Extraction and preprocessing of stock-related tweet data were handled by both Pranav and Adithya. The fine-tuning and training of the BERT and FinancialBERT models was a group effort, although model extraction and performance evaluation was primarily done by Pranav. Adithya played a key role in training and evaluating the AutoRegressive model. Finally, Adewale focused significantly on designing and implementing the confidence-based method for integrating predictions from the BERT-type models with those from the AutoRegressive model.

Github Repo

https://github.com/pranav-khetarpal/cse5525-final-project

References

Ran Aroussi. 2017. Yahooquery documentation.

Murtaza Dalal, Alexander C. Li, and Rohan Taori. 2019. Autoregressive models: What are they good for? *Preprint*, arXiv:1910.07737.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*.

A. R. Hazourli. 2022. Financialbert: A pretrained language model for financial text mining.

J. Lee, H. L. Youn, J. Poon, and S. C. Han. 2023. Stock-emotions: Discover investor emotions for financial sentiment analysis and multivariate time series.

V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi. 2016. Sentiment analysis of twitter data for predicting stock market movements. In *IEEE Conference on Computational Intelligence for Financial Engineering and Economics*. IEEE.

Christian Palomo. 2023. Tweet sentiment analysis to predict stock market.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

James H. Stock and Mark W. Watson. 2001. Vector autoregressions. *Journal of Economic Perspectives*.

zeroshot. 2024. Twitter financial news sentiment dataset.

A AutoRegressive Model Predictions on Test Set

Ticker	Directional Accuracy
AAPL	0.9191
AMT	0.8298
AMZN	0.8638
BA	0.8553
BAC	0.8809
BKNG	0.8766
BRK_B	0.8894
CCL	0.9021
CVX	0.8936
DIS	0.9106
GOOG	0.9021
GOOGL	0.9021
HD	0.9064
JNJ	0.8681
JPM	0.8426
KO	0.8766
LOW	0.9191
MA	0.8894
MCD	0.8894
MSFT	0.8511
NFLX	0.8128
NKE	0.8894
NVDA	0.9106
PFE	0.8638
PG	0.8894
SBUX	0.8766
TM	0.8596
TSM	0.8681
UNH	0.8596
UPS	0.8979
V	0.8128
WMT	0.8000
XOM	0.9064
ĜSPC	0.9149

Table 3: Directional accuracy of closing price per ticker on the test set: closing prices for the year of 2020.

B BERT + AutoRegressive Combined Approach Predictions on the Test Set

Ticker	Directional Accuracy
AAPL	0.8989
BA	0.8833
MSFT	0.8400
GOOG	0.5000
AMZN	0.9310
DIS	0.8966
CCL	0.9167
GOOGL	0.7500
NVDA	0.8750
SBUX	0.0000
MCD	1.0000
XOM	1.0000
V	0.7500
NFLX	0.8000
WMT	1.0000
NKE	1.0000
UPS	1.0000
JPM	1.0000
PFE	0.8824
UNH	1.0000
BAC	0.8571
HD	0.6667
KO	1.0000
MA	1.0000
JNJ	1.0000

Table 4: Directional accuracy of the combined model (BERT + AutoRegressive) per ticker, achieved using a Base Confidence of 0.65, a Confidence Threshold of 0.10, and a Boost Confidence of 0.10.

C FinancialBERT + AutoRegressive Combined Approach Predictions on the Test Set

Ticker	Directional Accuracy
AAPL	0.8989
BA	0.8833
MSFT	0.8400
GOOG	0.5000
AMZN	0.9310
DIS	0.8966
CCL	0.9167
GOOGL	0.7500
NVDA	0.8750
SBUX	0.0000
MCD	1.0000
XOM	1.0000
V	0.7500
NFLX	0.8000
WMT	1.0000
NKE	1.0000
UPS	1.0000
JPM	1.0000
PFE	0.8824
UNH	1.0000
BAC	0.8571
HD	0.6667
KO	1.0000
MA	1.0000
JNJ	1.0000

Table 5: Directional accuracy of the combined model (FinancialBERT + AutoRegressive) per ticker, achieved using a Base Confidence of 0.65, a Confidence Threshold of 0.10, and a Boost Confidence of 0.10.

D Combined Model Approach Pseudocode and Explanation of Algorithm

```
FUNCTION prediction_for_stock_and_day(auto_regression_direction, average_sentiment, base_confidence,
confidence_threshold, boost_confidence):

    // STEP 1: Convert sentiment score to binary direction
    // If sentiment is above 0.5, consider it bullish (1), otherwise bearish (0)
    IF average_sentiment > 0.5 THEN
        bert_direction = 1  // Bullish prediction from sentiment
    ELSE
        bert_direction = 0  // Bearish prediction from sentiment
    END IF

    // STEP 2: Adjust confidence based on agreement between models
    IF auto_regression_direction EQUALS bert_direction THEN
        // Models agree - boost confidence proportionally to sentiment strength
        // The further sentiment is from 0.5, the stronger the boost
        sentiment_strength = abs(average_sentiment - 0.5) * 2  // Normalizes to 0-1 range
        final_confidence = base_confidence + (boost_confidence * sentiment_strength)
    ELSE
        // Models disagree - reduce confidence proportionally to sentiment strength
        // The further sentiment is from 0.5, the stronger the reduction
        sentiment_strength = abs(average_sentiment - 0.5) * 2  // Normalizes to 0-1 range
        final_confidence = base_confidence - (boost_confidence * sentiment_strength)
    END IF

    // STEP 3: Initialize final prediction to auto-regression direction
    final_prediction = auto_regression_direction

    // STEP 4: If confidence falls below threshold, flip the prediction
    IF final_confidence < confidence_threshold THEN
        // Confidence too low, reverse the prediction
        IF final_prediction == 1 THEN
            final_prediction = 0
        ELSE
            final_prediction = 1
        END IF
    END IF

    // STEP 5: Return the final direction prediction
    RETURN final_prediction
END FUNCTION
```

Figure 1: Pseudocode for hybrid model integration logic

First, the continuous sentiment score (ranging from 0 to 1) is converted into a binary directional signal: 1 for bullish (up), and 0 for bearish (down). The model then compares this sentiment direction to the prediction made by the AutoRegressive (AR) model.

If both predictions agree, the model increases its confidence in the AR prediction. If they disagree, confidence is reduced, both adjustments are proportional to the strength of the sentiment, measured by how far the sentiment score deviates from the neutral midpoint of 0.5.

The default prediction is always the AR output. However, if the adjusted confidence falls below a defined threshold, the prediction is flipped. This flip indicates that the sentiment signal is strong enough to override the AR prediction.

Finally, the function returns a binary prediction (0 = down, 1 = up). This mechanism allows the model to adaptively weigh both prediction sources, giving more influence to sentiment only when it is strongly in disagreement with the time-series forecast.