Predicting Stock Price Movements Using Daily News

Shih-Lun Huang sh7008@nyu.edu
Hannah Park hp2501@nyu.edu
Jin Choi jkc9890@nyu.edu
George Zhou gz2214@nyu.edu

Abstract

This project compares different approaches to forecasting next-day stock price movements. It begins with a baseline model that uses LSTM networks with historical stock data as an input. Subsequently, Bidirectional Encoder Representations for Transformer (BERT) classification model is constructed. The final model combines LSTM network and BERT to predict stock prices by extracting embeddings from daily news headlines. Our model demonstrates improvement in the final model's predictive accuracy, with Mean Average Precision increasing by 0.11 to 0.56 and the AUC score rising by 0.16 to 0.59 compared to the baseline model. This highlights the value of integrating advanced machine learning techniques with news headlines for stock market forecasting.

1 Introduction

Stock price prediction is crucial in finance, and leveraging deep learning, particularly transformer models like BERT and GPT-2, shows great promise. Drawing inspiration from works such as [1] and [2], which highlight the predictive power of social media and financial news sentiment, our project aims to enhance stock price forecasts. We plan to develop a transformer-based pipeline to process news articles, creating embeddings to capture terminologies that may have impact on stock price prediction. These will be aligned with stock movement labels to predict price directions. Previous studies, using Twitter data, achieved 86.7% accuracy for the Dow Jones Industrial Average, indicating potential success with more relevant financial news. We anticipate challenges in gathering sufficient articles for quality embeddings. Our stretch goals include examining the impact of different news sources and integrating historical financial data like volatility measures, to further refine predictions.

2 Approach

In this project, we aimed to enhance stock price movement prediction by combining deep learning models with advanced natural language processing. Our objectives were:

1. Establish Baseline Model: Develop a baseline model using LSTM networks with historical stock data to set a standard for comparison.
2. Incorporate NLP Techniques: Integrate a pre-trained BERT classification model to analyze the impact of daily news headlines on stock market movements.
3. Develop Hybrid Model: Create a novel model that combines LSTM and BERT capabilities, leveraging LSTM for sequential stock data analysis and BERT for extracting meaningful insights from news headlines. Also conduct advanced analysis beyond sentiment by employing BERT to extract complex embeddings from news headlines, offering a more nuanced understanding of their impact on stock prices.

Project github repository: https://github.com/gz2214/Text2Trade

4. Quantitative Evaluation: Measure improvements in predictive accuracy using key metrics like Mean Average Precision and AUC scores, comparing the hybrid model against the baseline LSTM model.

The goal was to create a more accurate and comprehensive system for predicting stock market trends. By comparing and contrasting these models' performances, it allowed us to perform a comprehensive analysis of their predictive capabilities in the context of stock market forecasting.

3 Data and Preprocessing

3.1 Data

We gathered news article headlines from GNews, a simple, easy-to-use REST API that returns JSON search results for current and historic news articles published by over 60,000 worldwide sources. We specifically looked for articles in English relating to business and technology. We compiled a dataset spanning from 2020 to 2023 which contains more than 100,000 articles. Stock data was collected from the Yahoo Finance API, which provided historical data on stock prices. The date range of this stock price data aligned with the date range of news articles. We focused on the companies in the S&P 500, enabling us to comprehensively track the financial market without the need to exhaustively collect data from all listed firms.

3.2 Preprocessing

Target Labeling The label for the training data was derived from the daily stock price movement of S&P 500, specifically, the daily price movements are computed by deducting current opening price by the last market day opening price. Our model performed binary classification and predicted whether the stock price will go "up" or "down". The output results were compared to the actual stock price movements.

Daily News Aggregation In each section, a single day's data is represented by a row, which consists of a concatenation of all headlines from that day, along with a label indicating whether the S&P 500 stock price rose or fell. The target prediction is the stock's daily price movement, predicted using news from the previous day's start (9:30 am) to the start of the current day (9:30 am).

Handling Exception Cases of Holidays Additionally, when the dataset includes days when markets are closed, such as holidays, the analysis extends to cover the period from the last market open before the holiday to the next market open. The news articles published during these intervals are presumed to influence the stock price movement. For instances when the market is closed on holidays or weekends, there are no stock price movements on these specific dates. For example, with Christmas being a market holiday, the analysis would compare the stock price change from December 24th to December 26th, taking into account all articles published in this time frame.

Train and Test Split We trained and validated our results using blocked validation. The dataset was divided into contiguous blocks of time, and the model was trained on one block and tested on the subsequent block. By maintaining the temporal structure, this technique provided a more realistic assessment of a model's ability to generalize to unseen future data, helping to address issues such as data leakage and ensuring robust performance evaluation in the context of time-dependent datasets. Therefore, the raw dataset was segmented into five sections, with each section comprising 80% training data and 20% testing data.

Tokenization for BERT An issue we faced was the BERT model's limitation of 512 tokens. To handle this issue, the strategy was to split the day's news titles into 512 tokens, thereby having multiple news article chunks of size 512 tokens per day.

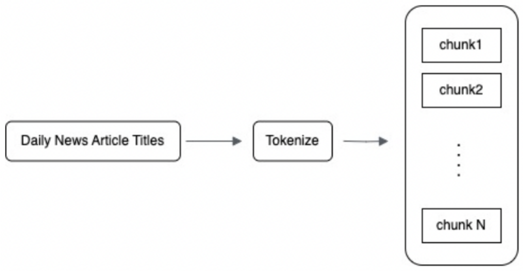

Figure 1: Tokenization of text into chunks

4 Experiments

We employed various models and techniques to achieve our goal of predicting stock price movements. We attempted three distinct approaches to compare model performances in stock price movement prediction.

4.1 LSTM Baseline Model

The baseline model employed LSTM with past stock price data as input for performance comparison. This allowed the evaluation of how adding news data impacts the model's performance in predicting stock price movements while also serving as a benchmark for comparing other models. To handle the sequential nature of stock prices, time-series techniques were applied by creating a dataset with rolling windows. Each data point included features from a specified lookback period, which was determined systematically to optimize precision based on LSTM model tuning results.

In addition, the training and tuning of the model were performed sequentially in a temporal manner. As outlined in 3.2, we split the price movement data into contiguous blocks. Rather than using the test sets in each block as test data, we utilized them for hyperparameter fine-tuning as validation sets. The last block was not included in the training phase and was instead reserved as a held-out test set. This sequential processing maintained the time-series structure of the data. The model had five hyperparameters, including the lookback period length, nodes, layers, dropout rate for LSTM regularization, and the learning rate of the optimizer. Bayesian optimization with the Optuna library was used to fine-tune these hyperparameters, efficiently exploring the space and focusing on areas likely to improve performance [3].

4.2 BERT for Sequence Classification Model

The second approach involved using a pre-trained BERT classification model to predict stock price movements as "up" or "down." A BERT for sequence classification model [4] was used to create this classifier. As previously mentioned, since the raw data was divided into 5 time-dependent blocks, and BERT models have a 512-token limit for input, the daily news headlines were split into 512-token chunks. Consequently, this resulted in multiple chunks for each day. The BERT model then processed these token chunks, and a Train-Test set split was conducted. hunks from block 0 to block 4's training set were used to train and fine-tuning the BERT classifier. Block 4's test set was used to evaluate the performance of the BERT classifier model.

After preprocessing and creating training and test datasets, the focus shifted to the classification task using the BERT for sequence classification model. Binary cross-entropy loss with a sigmoid output layer was used during training, and fine-tuning was done with the ADAM optimizer. To monitor the training process, training loss in each epoch was calculated and a total of 60 epochs were executed to train the model.

The final evaluation of the BERT model was performed on the test set, with a goal of generating aggregated daily results. We aimed to determine the likelihood of a stock price increase (class 1) and the corresponding classification (0 or 1) for each day. The evaluation process involved several steps. Since the number of news titles varied daily, resulting in different numbers of text chunks, the first step was to calculate the number of text chunks per day. This was accomplished by converting the test dataset into a dataframe and organizing the data by date to facilitate the evaluation process.

During the evaluation, both the logits and probabilities for class 1 were stored when the model generated predictions for each chunk of news title text. To derive the final probabilities for each day, we employed an aggregation method that calculated the mean probability across all text chunks for that specific day. The final predictions for each day were determined through a majority vote across all chunks for the specific day. The individual day probabilities and predictions were consolidated and structured into appropriate tensors.

4.3 LSTM with BERT Embeddings

The process for analyzing daily news using BERT involved tokenizing news headlines and breaking them into smaller chunks to fit within BERT's token limit. Each chunk was transformed into embeddings, representing the sentiment and content of the news. This was necessary because BERT can only process a maximum of 512 tokens at a time.

After tokenizing headlines with a BertTokenizer, a custom function split these tokens into manageable chunks. These chunks were then fed into the BERT model to create embeddings (with dimensions reflecting tokens and BERT's hidden layer nodes). These embeddings were averaged to form a context vector for each chunk.

To encapsulate a day's news, the process was repeated for each chunk. The resulting embeddings were then averaged to produce a single vector, representing the day's news content and sentiment. Finally, these daily news embeddings were integrated with daily price movement data, undergoing the same processing, training, tuning, and evaluation steps as the baseline model.

5 Analysis

To evaluate the comparative performance of the models, we employed several key evaluation metrics, such as accuracy, precision, F1 score, and AUC score. These metrics collectively provide a comprehensive assessment of the model's effectiveness and overall performance.

Baseline LSTM Model After hyperparameter tuning, the best model was composed of 3 layers with 80 nodes each and a dropout rate of 0.42. Additionally, the lookback period length was 30 days. This setup yields the result of 0.43 for AUC score, 0.5 for accuracy, 0.5 for precision, and 0.516 for F1 score as displayed in Table 1.

BERT for Sequence Classification Model As the BERT classification model outputs labels directly, we couldn't calculate an AUC score, which needs logit values. Upon analysis, we observed a precision value of 0.476, an F1 score of 0.540, and an accuracy score of 0.468. The results of our evaluation indicate that the model's performance falls slightly below what would be expected by random chance or is roughly equivalent to random chance.

LSTM with BERT Embeddings The embedding vector of daily news headlines generated from BERT contained 1,536 features, and since we aggregated the chunked inputs into daily-level representation, it resulted in dense embedding vectors. Tuning the LSTM model on this new set of inputs, we obtained an LSTM model constructed by 2 layers with 30 nodes each, using a 30 days lookback period. This configuration achieved an AUC score of 0.59, with precision of 0.625, accuracy of 0.567, and an F1 score of 0.435, as detailed in Table 1.

Discussion From Table 1, we see a mixed result for both BERT Classifier model and LSTM with BERT Embeddings when compared to the baseline model. We first investigated our data distribution and noticed that it is fairly balanced with a 46% downward movement, hence, metrics like AUC score and accuracy which are sensitive to imbalance data could provide a valid evaluation of the models. From the results, we see that BERT Classifier wasn't able to surpass the baseline model despite a better F1 score, which only indicates a good balance between precision and recall. Conversely, LSTM with BERT Embeddings model outperformed the baseline model in most metrics except F1 score. This suggests its general effectiveness but also highlights a need for improved balance in handling false positives and negatives, which are crucial in stock price movement forecasting.

6 Conclusion

In this project we explored the potential of utilizing NLP techniques, specifically the BERT model, for financial tasks. Initially, the BERT Classifier alone didn't show a strong link between news headlines and stock prices. However, combining news with past price data improved the model's ability to assess news impact on prices, a novel approach compared to traditional NLP applications in finance which utilized an extra process of sentiment score analysis.

The potential of the LSTM with BERT Embeddings model is clear, yet improvement is needed. Achieving a finer balance between precision and recall is key to minimizing risks associated with overly optimistic predictions and missed bullish opportunities. Our study, limited to 797 trading days and excluding data like volatility and trading volumes, paves the way for future enhancements. We plan to expand our dataset and conduct a comparative analysis using an LSTM model enhanced with sentiment analysis. This will enable us to refine our methodology and assess its performance relative to established financial forecasting models.

Table 1: Comparison of Model Performance

Model	**AUC**	**Precision**	**Accuracy**	**F1**
Baseline	0.43	0.5	0.5	0.516
BERT Classifier	-	0.476	0.469	0.541
LSTM with BERT Embeddings	0.59	0.625	0.567	0.435

Contribution

George was responsible for handling data collection through APIs and performing data preprocessing. Jin developed functions to split text data into chunks that fit within the token limit of the BERT model, as well as computing word embeddings for the BERT model. Hannah was tasked with constructing the BERT classifier, our second model in the project. Shih-Lun built the LSTM model and was responsible for running both the baseline and proposed models.The entire team collaborated in writing the report.

References

[1] Mao Huina Bollen, Johan and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science, vol. 2, no. 1, Mar. 2011, pp. 1–8. ScienceDirect*, 2011.

[2] Qianyi Xiao and Baha Ihnaini. Stock trend prediction using sentiment analysis. *PeerJ Computer Science, vol. 9, Mar. 2023, p. e1293. PubMed Central*, 2023.

[3] Larochelle Hugo Snoek, Jasper and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. advances in neural information processing systems. *Advances in neural information processing systems, 25*, 2012.

[4] Huggingface. Training and fine-tuning. https://huggingface.co/transformers/v3.3.1/training.html.