**Financial Text Analysis: Using NLP to Forecast Stock Prices**

**Thanawan Lertmongkolnam, Neeharika Goyal, Vorapoom Thirapatarapong,**

**Rishitha Golla, Paniti Mongkonpathumrat**

University of Southern California

(lertmong, neeharik, thirapat, rgolla, mongkonp) @usc.edu

**Abstract**

Several machine learning models have been used to analyze historical financial data and predict the stock price to support short-term trading strategies. According to the efficient market hypothesis, the current stock price reflects all information in the market, and if new information becomes available, the price will respond accordingly. This means that not only numerical financial data but other forms of information such as news, articles, and financial documents can affect the stock as well. This paper aims to address the benefits of integrating fundamental analysis with technical analysis. Fundamental analysis typically requires a thorough manual examination of financial documents. We have included annual reports (10-K), quarterly reports (10-Q), and current reports (8-K) of 10 largest stocks in Dow Jones with 5-year historical data from SEC’s EDGAR system. Our strategy utilizes NLP to extract relevant features from these financial records and integrate it with the baseline model which solely depends on technical analysis. Our preliminary results show that we achieved a slight improvement in stock price prediction when fundamental analysis is considered.

**1 Introduction**

Stock prices are a crucial element of the financial market, serving as an indicator of the performance of companies and the economy as a whole. The importance of stock predictability is further heightened in today's highly dynamic and complex financial market. It is essential as it helps investors make informed decisions about buying, selling, or holding stocks. It is essential as it helps investors make informed decisions about buying, selling, or holding stocks.

Research suggests that performing sentiment analysis on stock market data can aid investors in making informed decisions based on market trends. We believe that by integrating fundamental and technical analysis, our proposed model may improve stock market forecast performance.

**2 Related Works**

Previously, machine learning models were used to process historical financial data and textual data separately, and the interconnection between numerical and textual information that could affect the stock price is overlooked. Recently, the joint effect of various types of information has been studied. In Kalra (2019), to predict the Indian stock market, news articles and historical data were used. News articles were preprocessed using tokenization, transformation, stop word removal, Term Frequency, and Inverse Document Frequency (TF-IDF). Sentiment analysis has been performed on the preprocessed dataset by the Naive Bayes Classifier. Then, sentiment values and historical data were integrated by date to form predictors and those predictors were given to the Machine Learning models such as KNN, Neural network, Super Vector Machine, and Naive Bayes for stock prediction.

In Guo (2020), news articles and price history were used to predict stock prices. Sentiment analysis has been done on the textual data (news) using two python libraries, VADER (Valence Aware Dictionary and Sentiment Reasoner) and TextBlob, which changes text into sentiment scores. After that, all the essential features from news and historical stock basic features have been used in the LSTM neural network to predict close prices and stock returns.

For this project, financial documents will be used instead of news articles, and multiple features extractions techniques, such as FINBERT, Flair, VADER and TextBlob will be used and compared to see which will work best. Moreover, we will use machine learning models such as LSTM and GRU to predict the stock price to see if including information from financial documents will improve the model's performance.

**3 Problem Description**

From the facts that integrating newly-available information and analyzing fundamental information about the corporates from the financial reports published can improve the stock price prediction, we expected that adding information from financial reports has a potential to help with the forecast accuracy. Thus, the objective of this project is to study whether using the features extracted from financial documents along with technical analysis can help improve the model performance compared to a model that only uses technical analysis.

In order to do so, the project was broken down into three main parts. First, the financial reports data were collected from SEC’s EDGAR system, and the contents in these reports were extracted into sentiment values features by using different NLP techniques. Second, the baseline model that only uses technical analysis was developed. Finally, the features extracted from the financial reports were then integrated into the baseline model.

Due to time and resource limitations, the project was scoped on the data from the last five years (2018 to 2023) for the 10 stocks with the highest market caps listed on Dow Jones, including AAPL, MSFT, V, UNH, JPM, JNJ, WMT, PG, CVX, and HD.

**4 Methods**

**4.1 Extracting Sentiment Features from Financial Reports using NLP tools**

We have investigated the financial reports of 10 largest stocks in Dow Jones (as of March 8 2023) with 5-year historical data. Three types of reports that we used include Current Events report (8-K), Quarterly report (10-Q), and Annual report (10-K). These will be used as additional features for stock price prediction. From the aspect of fundamental analysis in stock prediction, we considered sections in the report that have potential to improve the stock price prediction. For 8-K, we include every section because this document is designed to inform significant items to the public that can affect investment decisions. For 10-Q, only part I: item 2, 3 are included. For 10-K, only part I: item 1A, 2, 3, part II: item 5, 7A are included.

The content of the financial report was then extracted using four different NLP Sentiment Analysis tools; VADER (Valence Aware Dictionary and Sentiment Reasoner), TextBlob, FinBERT, and Flair. These extracted features will be used as additional features for the stock price prediction model in the next step. Details of each tools are as follows:

1. VADER, a rule-based sentiment analysis tool, generates 4 features: pos, neu, neg, and compound. The first three features represent the proportion of the text that falls into those categories. The compound feature is the normalized sum of all of the lexicon ratings (Bajaj, 2021).
2. TextBlob, another lexicon-based approach, returns two values: polarity and subjectivity (quantifies the amount of personal opinion and factual information) scores (Shah, 2020).
3. FinBERT, a financial version of BERT, returns three probability scores of each sentiment class: positive, negative, and neutral (Goncharov, 2022).
4. Flair, a state-of-art NLP framework built on PyTorch, outputs both predicted label and sentiment scores ranging from -1 to 1 (Shahul, 2023).

**4.2 Developing Baseline Model for Stock Price Prediction using Only Financial Data**

Our baseline model for stock price prediction is a specific architecture of the Long Short-Term Memory (LSTM) model. The input features include 10 days of historical stock price information, (Open, High, Low, Close, and Volume), and Dow Jone Index (which is used as a control variable to capture the market movement). The target variable is the Close price for that particular day. These stock price data were obtained from Python’s library yfinance.

The ratio of training and testing datasets is 80:20. We used an Adam optimizer and MSE loss function. Our optimal parameters were the learning rate of 0.001 and 200 epochs of training. RMSE (Root Mean Square Errors) and SI (Scatter Index) were used to evaluate the performance.

**4.3 Adding the Sentiment Features from Financial Reports to the Baseline Model**

In the initial approach, each category of the sentiment features was employed in such a way that the three reports were handled individually. For example, we added Flair sentiment scores from 8-K, 10-Q, and 10-K as three different features.

We merged the sentiment features with the financial features using dates and filled the blanks using the most recent information, known as forward fill. Each category of the sentiment features was used to train the model independently for comparison.

Then, the improvement on the first version of the combined model was done in two aspects: using feature engineering and choosing another model.

**4.3.1 Improving the Performance using Feature Engineering**

Three variations on the feature engineering were explored.

1. 10-day ffill: Instead of using forward fill, we limited the fill to only 10 days after the most recent information for each sentiment feature. This is due to the assumption that the effect of the financial reports on stock price only lasts for a period of time.
2. Mask features: We added mask features whose values are either 1 or 0 to indicate whether that particular data point has the values for the sentiment features or not, then padded the blanks in the sentiment features with 0. The purpose was to distinguish them from the data points where the values of the sentiment features are actually 0.
3. Combining 8-K, 10-Q, and 10-K: Instead of handling the three reports individually, we merged the same sentiment features from 8-K, 10-Q and 10-K together based on the assumption that all types of reports should affect the stock price similarly. For example, instead of having Flair sentiment scores for 8-K, 10-Q, and 10-K, we have only one Flair feature representing all financial statements.

**4.3.2 Improving the Performance using Gated Recurrent Unit (GRU) model**

GRU was implemented with the third type of feature engineering, called combined features, to evaluate if a different model will help improve the prediction result.

For the model setup, we used a Root Mean Square propagation (RMSprop) optimizer and MSE loss function. Our optimal parameters were the hidden dimension of 64, the learning rate of 0.0001, and 180 epochs of training.

| **Stock** | **Base**  **line** | **VADER** | **Text**  **Blob** | **Fin**  **BERT** | **Flair** |
| --- | --- | --- | --- | --- | --- |
| AAPL | 2.45% | 2.48% | 2.70% | 4.45% | 2.39% |
| MSFT | 2.50% | 2.19% | 2.30% | 2.39% | 2.19% |
| V | 1.59% | 1.75% | 1.58% | 1.80% | 1.62% |
| UNH | 2.08% | 2.37% | 2.19% | 2.06% | 1.94% |
| JPM | 1.83% | 1.71% | 2.01% | 1.95% | 2.23% |
| JNJ | 1.17% | 1.34% | 1.39% | 1.23% | 1.49% |
| WMT | 1.32% | 1.26% | 1.39% | 1.27% | 1.26% |
| PG | 1.39% | 1.75% | 2.61% | 1.89% | 1.49% |
| CVX | 2.31% | 3.08% | 2.48% | 3.51% | 3.04% |
| HD | 2.23% | 2.13% | 2.26% | 2.30% | 2.15% |
| **Avg.** | 1.89% | 2.01% | 2.09% | 2.29% | 1.98% |

Table 1: Comparison of Scatter Index (SI) values from LSTM baseline model and models with different sentiment features using the initial approach.

| **Method** | **Baseline** | **VADER** | **TextBlob** | **FinBERT** | **Flair** |
| --- | --- | --- | --- | --- | --- |
| Initial | 1.89% | 2.01% | 2.09% | 2.29% | 1.98% |
| 10-day ffill | 2.06% | 2.21% | 2.25% | 2.09% |
| Mask features | 2.13% | 2.04% | 2.02% | 1.98% |
| Combined | 1.95% | 1.86% | 1.87% | 1.93% |

Table 2: Comparison of Scatter Index (SI) values of LSTM models with sentiment features from financial reports with different versions of improvement on feature engineering.

**5** **Experimental Results**

**5.1 Baseline model with and without financial reports features**

According to Table 1, the LSTM baseline model achieved the average SI value of 1.89% from the 10 stocks, ranging from 1.17% to 2.50%.

After adding each feature extracted from financial reports with the initial approach, there was no improvement shown on all types of features as SI values were higher than the baseline.

**5.2 Model Improvement using Feature Engineering in LSTM**

After determining that the original strategy did not aid in stock price prediction, the attempts of improvement were carried out by experimenting with various feature engineering methodologies. Table 2 shows that the first two techniques, 10-day ffill and adding mask features, had no positive effect on model performance when compared to the baseline. However, when employing TextBlob and FinBERT characteristics, the third technique, which combined features from all three report types into one feature, showed a little improvement, with SI values decreasing to 1.86% and 1.87%, respectively.

**5.3 Model Improvement using GRU (Gated Recurrent Unit) model**

Apart from using feature engineering methods to improve the model performance, we also attempted to use a GRU-based model with combined features to see if it helps with this specific task of stock price prediction. According to Table 3, we observed a slight improvement in prediction when TextBlob features were used as the SI values decreased from 2.32% to 2.27%.

| **Stock** | **Base**  **line** | **VADER** | **Text**  **Blob** | **Fin**  **BERT** | **Flair** |
| --- | --- | --- | --- | --- | --- |
| AAPL | 3.38% | 3.47% | 3.23% | 3.57% | 3.22% |
| MSFT | 3.16% | 2.93% | 2.98% | 3.32% | 3.02% |
| V | 2.10% | 2.05% | 2.05% | 2.30% | 2.08% |
| UNH | 1.96% | 1.87% | 1.97% | 2.53% | 1.91% |
| JPM | 2.41% | 2.51% | 2.36% | 2.38% | 2.85% |
| JNJ | 1.13% | 1.21% | 1.10% | 1.19% | 1.48% |
| WMT | 2.63% | 2.58% | 2.54% | 2.54% | 2.58% |
| PG | 4.67% | 5.04% | 4.65% | 5.12% | 4.75% |
| CVX | 1.14% | 1.28% | 1.14% | 1.16% | 1.36% |
| HD | 0.63% | 0.63% | 0.64% | 0.63% | 0.62% |
| **Avg** | 2.32% | 2.36% | 2.27% | 2.47% | 2.39% |

Table 3: Comparison of Scatter Index (SI) values from GRU-based model with and without sentiment features using the combined feature

**6 Conclusion and Future Work**

In this paper, we explored the stock price prediction models that integrate technical analysis with fundamental analysis using NLP techniques to extract relevant features from financial reports. With several variations of feature engineering and two different algorithms, we found that TextBlob has slightly improved the performance of both LSTM and GRU and FinBert has slightly improved the performance of LSTM as well.

There are several potential improvements that could be done for a future work: 1) expanding the scope in terms of number of stocks and duration 2) gathering more historical financial information such as earning-per-share, revenue, etc. 3) exploring different NLP techniques to extract features that can better interpret the sentiments of the reports.

**7 Individual contribution**

| **Tasks** | **Contribution** |
| --- | --- |
| Financial reports investigation | * Thanawan Lertmongkolnam * Neeharika Goyal * Vorapoom Thirapatarapong |
| Feature extraction using different NLP techniques |
| Developing baseline models | * Rishitha Golla * Paniti Mongkonpathumrat |
| Develop and tune combined models | * Thanawan Lertmongkolnam * Paniti Mongkonpathumrat * Rishitha Golla * Vorapoom Thirapatarapong |
| Final report | * Neeharika Goyal |

**8 References**

* Aryan Bajaj. 2021. Can Python understand human feelings through words? – A brief intro to NLP and VADER Sentiment Analysis. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/vader-for-sentiment-analysis/>
* Khan, R. (2018). LSTM for Google Stock Price Prediction. Towards Data Science.<https://towardsdatascience.com/lstm-for-google-stock-price-prediction-e35f5cc84165>.
* Parthvi Shah. 2020. Sentiment Analysis using TextBlob. Toward Data Science. <https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524>
* Ivan Goncharov. 2022. Financial Sentiment Analysis on Stock Market Headlines With FinBERT & HuggingFace. Weights & Biases. <https://wandb.ai/ivangoncharov/FinBERT_Sentiment_Analysis_Project/reports/Financial-Sentiment-Analysis-on-Stock-Market-Headlines-With-FinBERT-Hugging-Face--VmlldzoxMDQ4NjM0>
* Shahul ES. 2023. Sentiment Analysis in Python: TextBlob vs Vader Sentiment vs Flair vs Building It From Scratch. Neptune.ai. <https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair>
* TensorFlow. (n.d.). tf.keras.layers.LSTM. TensorFlow API documentation.<https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM>.
* Ya Gao, Rong Wang, and Enmin Zhou. 2021. Stock Prediction based on optimized LSTM and GRU models. <https://doi.org/10.1155/2021/4055281>