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

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

**Rishitha Golla, Paniti Mongkonpathumrat**

**1 Tasks that have been performed**

* We have investigated the financial reports from SEC’s EDGAR system which will be used as additional features for stock price prediction. Three types of reports include Current Events report (8-K), Quarterly report (10-Q), and Annual report (10-K). From 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 decision.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. We have included data for 8 out of the 10 largest stocks in Dow Jones that we aim to cover.
* We attempted to develop a code to crawl the SEC website in order to access the financial reports of different firms using both Java and Python. Unfortunately, since the structure of the web pages was not consistent and had changed HTML tags hierarchy, writing a uniform code to crawl the web page to extract the Financial Report made the task challenging. Further the structure of the reports varies significantly which makes it difficult to collect the data in a structured manner. Therefore, we decided to download the financial report data manually to better serve our project’s purpose.
* 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:
  + 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).
  + TextBlob, another lexicon-based approach, returns two values: polarity and subjectivity (quantifies the amount of personal opinion and factual information) scores (Shah, 2020).
  + FinBERT, a financial version of BERT, returns three probability scores of each sentiment class: positive, negative, and neutral (Goncharov, 2022).
  + Flair, a state-of-art NLP framework built on PyTorch, outputs both predicted label and sentiment scores ranging from -1 to 1 (Shahul, 2023)
* We have developed some baseline models for stock prediction, including Gated Recurrent Units (GRU) networks(Gao, 2021), Long Short-Term Memory (LSTM) networks, and Feedforward Neural Networks (FNN).
* We used Yahoo finance API to extract five years of historical stock data. We have applied a 10 day lag to create features.
* We used classification and regression models to predict the stock prices for baseline models. LSTM and FNN models were used for classification tasks to predict whether the stock price increased or decreased, while GRU networks were used for regression tasks to predict the actual stock price value.
* In terms of performance, FNN had a higher precision of 51% compared to LSTM’s 48%. Both FNN and LSTM had high recall values of 100%. Overall, the baseline model of FNN performed better than LSTM when comparing precision and recall. For the GRU regression model, performance is evaluated using root mean squared error (RMSE). The GRU model had a RMSE of 0.83.

**2 Risks and challenges**

* The structure of the financial reports of different companies varies significantly in terms of both report length and languages used. Therefore, this could lead to biases of the sentiment values extracted from different companies due to these discrepancies.
* The amount of data available in financial reports is overwhelming, and it becomes difficult to determine which data to include that would be relevant for stock price prediction models. The reports contain a large amount of data in various formats, such as tables, charts, and text, making it difficult to maintain consistency among data for use as features in stock price prediction models.
* Kernel crashed when running FinBert.
* Small dataset: Due to time and resource limitations, we chose to focus the effort on data from the last five years. Also, obtaining stock price data at a frequency higher than daily is expensive, which has resulted in a small number of instances in our dataset. This issue is a main challenge for models that require a huge dataset, such as neural networks.
* Volatile stock price: Stock prices are highly volatile, so it is difficult to significantly increase the prediction accuracy.

**3 Plan to mitigate the risks and address the challenges**

* Instead of directly using the sentiment values extracted from the financial statements, we plan to normalize these values in order to mitigate the potential bias.
* Applying necessary data cleaning and preprocessing techniques to ensure consistency among data in various formats into a standardized format, such as a numerical or text-based format.
* Data cleaning and debugging will be performed to mitigate problems with kernel crashes in FinBert. If it does not work, new tools will be explored.
* Instead of solely focusing on neural networks, we plan to also consider some other models that do not require huge data size, such as Autoregressive moving average (ARMA) model, Autoregressive integrated moving average (ARIMA) model, and Seasonal autoregressive integrated moving average (SARIMA) model.

**4 Individual contribution**

| Group Member | Contribution |
| --- | --- |
| * Thanawan Lertmongkolnam * Neeharika Goyal * Vorapoom Thirapatarapong | Financial reports investigation, and feature extraction using different NLP techniques |
| * Rishitha Golla * Paniti Mongkonpathumrat | Developing baseline models |

**5 Links to existing work**

* [Feature extraction from financial reports](https://drive.google.com/drive/folders/1TmH5HVpAF6xF1Jz_62JmmW536WGkN_Fg)
* [Baseline models](https://drive.google.com/drive/folders/1n2lwF7xFm65Cy_kP9HViY4n0nD6IvHw2?usp=share_link)

**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>