Investigating the Impact of Long-Form Financial News Articles on Stock Market Fluctuations

CS 485: Applications of Natural Language Processing

Arnav Kolli, Anvesh Sunkara, Rahul Vedula

1. Abstract

This paper investigates the relationship between financial news and stock market fluctuations by attempting to gauge the impact of an article's content and sentiment on the opening price of a stock the following day. To do this, we developed a multitude of NLP models trained on a dataset comprising of article content, stock tickers, and stock prices. The results of our investigation showed that an article's impact on a stock, while tangible, is hard to predict (as evidenced by low accuracies in our models) due to the extremely nuanced nature of the stock market.

2. Introduction

Attempting to predict stock market fluctuations is something that, if done correctly, can have huge fiscal upside. This has led to the creation and development of models, companies, and even industries dedicated to predicting and anticipating these fluctuations. These predictions are usually based on either technical analysis, which analyzes historical data and chart patterns to identify suitable investment points, or fundamental analysis, which focuses on measuring an asset's intrinsic value through economic, financial, and other qualitative/quantitative factors.

While it is said that short-term traders prefer technical analysis and long-term value investors fundamental analysis, given the rise of algorithmic trading and the vast amount of fast-disseminating information, we wanted to explore the potential of qualitative fundamental analysis - through news content and sentiment - in short-term trading. To that end, this paper attempted to answer the following research question: Does a tangible relationship exist between the content and sentiment of a news article and the opening price of the associated stock the day after the article was published? Further, can this relationship be used to accurately predict the movement of stock prices?

To accomplish this, we decided to develop and evaluate the performance of multiple NLP models trained on a dataset we assembled consisting of information such as stock prices, article content, and headlines. Each model will analyze the same dataset pertaining to major companies, aiming to predict the direction and magnitude of stock price changes. Our intention in researching various models is to create a comprehensive assessment of the most accurate models for predicting stock price fluctuations compared to evaluating a single model's effectiveness.

Specific, additional research goals include:

- Determining whether the sentiment expressed in financial articles can accurately forecast stock price trends for major companies.

- Comparing the accuracy of various NLP models against each other in predicting stock movements to find a model that would help best with objective one.
- Identifying keywords or phrases in the articles that have a strong correlation with stock price shifts.

3. **Related Work**

Exploring stock market fluctuations through the qualitative lens of news analysis is a topic that considerable research has been performed on. This section reviews existing research that was influential and impactful in conducting our own research on stock-market predictions.

Two studies in particular served as foundations for our research: The first was a research investigation conducted by Robert P.Schumaker and Hsinchun Chen and is titled "A quantitative stock prediction system based on financial news" (Robert P. Schumaker a et al. A quantitative stock prediction system based on Financial News). This research study aimed at using a synthesis of linguistic, financial, and statistical techniques to create the Arizona Financial Text System which was used to predict stock price based on financial news articles. The second study, by J. Bollen, H. Mao, and X. Zeng, is titled "Twitter mood predicts the stock market" (insert citation) and delved into the correlation between mood states derived from large-scale Twitter feeds and the value of the Dow Jones Industrial Average over time. These sources served to supplement our own investigation by providing us with research techniques and methodologies that we used to create our dataset and models.

4. **Data Collection**

Initially, our dataset was to be a dataset on Kaggle that contained over a million article headlines across 6000 distinct stock tickers. We wanted to be more specific in the stocks that we chose and decided to limit ourselves to 2023's list of Fortune 50 companies. However, when we ran our first iteration on this list, we had a startlingly low number of articles due to the dataset's information being from a couple years ago and some companies on the Fortune 50 list not being publicly traded. We therefore decided to sort the dataset by ticker according to the number of headlines. This was the list of tickers that we ended up with:

['MRK', 'AMZN', 'MS', 'MU', 'NVDA', 'QQQ', 'M', 'EBAY', 'NFLX', 'GILD', 'DAL', 'JNJ', 'QCOM', 'BABA', 'KO', 'ORCL', 'FDX', 'HD', 'BB', 'BMY', 'JCP', 'LLY', 'CMG', 'CAT', 'GPRO', 'CHK', 'FSLR', 'NOK', 'P', 'LMT', 'MCD', 'MA', 'EA', 'FCX', 'GPS', 'PEP', 'GRPN', 'HAL', 'LOW', 'ADBE', 'AZN', 'MYL', 'DISH', 'ATVI', 'MDT', 'DB', 'LNKD', 'AA', 'EWU', 'AGN', 'EWJ', 'GLD', 'EWP', 'EWC', 'APC', 'AVGO', 'PCLN', 'AIG', 'EWZ', 'GOOGL', 'CCL', 'HUM', 'FCAU', 'DD', 'CRM', 'MMM', 'BBRY', 'BIIB', 'EWI', 'BIDU', 'DE', 'AXP', 'CMCSA', 'CVS', 'PFE', 'KR']

As our idea changed, we decided to test our models **not just on headlines, but also** on long-form texts, i.e, article body content. This meant that the dataset on Kaggle wasn't going to be enough as it contained just headlines and a URL to the body, but not the body itself. We therefore had to find a workaround to procure our financial news.

4.1. **Financial News.** The first thing we tried was scraping the URL links present within the dataset. This worked initially, but as we continued scraping we hit paywalls on the website that the URLs linked to. It was at this point that we discovered Benzinga's financial news API. We were able to create a trial account and access the API, however we still needed to fetch the content. Having seen the article bodies for some of the headlines, we discovered that while the dataset had an associated ticker with each headline, a lot of these articles mentioned the ticker in passing and in conjunction with many other stocks. An example of such an article is "Stocks That Hit 52-Week Highs On Tuesday" whose associated ticker was ADBE (Adobe). Such articles were useless to us in that they barely even mentioned the company in question. We, therefore, decided to filter the Kaggle dataset to only include headlines that mentioned the company by name within the headline. This was done by creating a secondary data structure that contained possible names for the company and looked like this:

'LOW': ["Lowe's", 'Lowe's Companies'], 'ADBE': ['Adobe Inc.', 'Adobe']

Once we narrowed our headlines down, we iterated through them and fetched the body for each of these articles from the Benzinga API using the headline as a search parameter.

4.2. **Stock Prices and Annotations.** To fetch the opening prices for the given stock on the day that the article was published and the next day, we used an unofficial Yahoo Finance API (the original one was discontinued). This was a simple endeavor for the most part as all we had to do was iterate through our current dataset and fetch the prices for the two days.

Note: For the remainder of this exploration, the day of the article being published will be referred to as the first day while the next day will be referred to as the second day.

The nuances when it came to fetching stock prices (the day of article being published and the subsequent day respectively) came into play when either the first or the second day happened to be a weekend when the stock market is closed. When this happened, we fetched the prices differently in the following manner:

- If the first day happened to be a weekend, we fetched the price for the Friday before the weekend. Essentially:

```
if(Day_one in ("Saturday", "Sunday")):
    Day_one_price = fetch opening_stock_price("Friday")
```

- If the second day happened to be a weekend, we fetched the price for the Monday after the weekend. Essentially:

```
if(Day_two in ("Saturday", "Sunday")):
    Day_two_price = fetch opening_stock_price("Monday")
```

This gave us a final dataset that looked like this:

The final dataset had 17,021 unique article bodies across 60 different tickers. The label is our annotation for that stock that shall be explained further in the Methodology section.

TABLE 1. Snippet of Dataset

Headline	Article Date	Ticker	Open Price	Open Price_	Article Body	Label
Alibaba Invests...	5/20/20	BABA	220	211.289993	Alibaba Group Holding Ltd NYSEBABA announces...	D
Here's How Much...	5/19/20	BABA	216.729996	220	Investors who owned stocks in the past five years...	I
'Fast Money' Pick...	4/27/20	BABA	207.550003	204.809998	On CNBCs Fast Money Half-time Report Stephanie...	D
...	...	...	...	...	...	...

5. **Methodology**

5.1. **Classification Model.** To start off, we initially attempted to predict the fluctuations by percentage instead of a classifier model. This proved to be extremely difficult which led to us switching to a multi-class classification model. We had three classes: Increase (I), Decrease (D), and Neutral (N). Given the extremely low probability that any stock might open at the *exact* same price as the previous day, we decided to have the criteria for the Neutral category be a price movement within $\pm 1\%$ of day one's opening price. We settled on 1% because it is the average daily movement of the S&P 500 stock and we wanted to see if a news article could cause an out-of-the-norm price movement.

5.2. **Data Engineering and Feature Engineering.** During our investigation, we experimented with pre-processing our data in the form of stopword removal, tokenization, and even stemming. Numerical data within the texts was not handled in a specific way nor did we factor any of the data into our models. This was because we only wanted to analyze the impact of text and sentiment as compared to quantifiable data.

5.3. **Models Tested.** For the purpose of this research exploration, we decided to evaluate the performance of multiple models instead of testing just one. The models we chose for this exploration were:

- Support Vector Machine (SVM)
- Convolutional Neural Network (CNN)
- Random Forest
- Bidirectional Encoder Representations from Transformers
- NLTK Sentiment Analysis

5.3.1. *Support Vector Machine.* We chose to use an SVM as a part of this exploration as SVMs are well-suited for multiclass classification tasks. They are also known to be ideal for handling datasets with high dimensionality. Given the features of our dataset and financial data not being linearly separable, it made sense to evaluate the performance of SVM on our dataset. For this model we used the following specifications: We used a TF-IDF vectorizer with max 5000 max features and default hyperparameter settings. The random state was 42 and the test_size 0.2.

5.3.2. *Convolutional Neural Network.* While CNNs are commonly used for image analysis, they are also known to be effective for text data by capturing local patterns. This can prove to be useful for analyzing long-form news articles and the sentiment within them. This is because CNNs establish a hierarchy of features starting from low-level features like words scaling to mid-level features such as syntactic structures to high-level features that capture the overall context. For this model, we used a standard Keras tokenizer without specified hyperparameters and padded sentences to the maximum length found in the dataset. The CNN layers in this model were: 'Embedding', 'Conv1D', 'GlobalMaxPooling1D', 'Dense', 'Dropout' with default hyperparameters. During the training process of this model, we had issues with the model overfitting the training data which resulted in an

5.3.3. *Random Forest.* A Random Forest model use multiple decision trees to make predictions for classification tasks. They can be used to identify influential words that correlate to stock-price fluctuations in our news articles. They're also better at handling overfitting and noisy data that is rampant in news articles. Default hyperparameters were used for this model.

5.3.4. *Bidirectional Encoder Representations from Transformers.* The BERT is a model that is used to deep-analyze the context of a text by passing the input through multiple layers. This makes it extremely useful for our exploration as it could be used to understand the relationships and context between words leading to a thorough sentiment analysis. We used the 'TFBertForSequenceClassification' from the Hugging Face Transformers library with a learning rate of 5e-5 and a categorical cross-entropy loss function with from_logits = True. This indicates that the output of the model was not normalized. The model was trained for three epochs with a batch size of 8.

5.3.5. *NLTK Sentiment Analysis.* The NLTK library's sentiment analysis served as our control group. We used it as a baseline comparison against our other models.

5.4. **Experimental Setup.** Given the sheer size of our dataset and the amount of content we had to parse, the computation power required to perform this investigation was pretty high. This meant that we had to use a GPU as pure CPU would have caused the models to run for an extremely long time. We therefore Google Colab's T4 GPU Hardware Accelerator to run all the models (coded in Python3).

6. **Evalutation**

Once the models were trained, we evaluated each of them separately and then compared the results with each other. Evaluating different models allowed us to come up with various insights that were extremely useful for drawing conclusions. The purpose of our evaluation

was also to assess how reliable our models were for real-life application and usage. We used a combination of evaluation metrics for each model to evaluate their performance. The chosen metrics were:

- Accuracy
- F-1 Score
- Precision
- Recall
- Loss Functions

6.1. **Accuracy.** The accuracy of an NLP model is defined as the ratio of correct predictions to total predictions within the dataset. A higher accuracy implies that the model was more successful in predicting the labels for a particular data point while a lower accuracy implies that the model was not as successful.
We calculated the accuracies for our model and plotted them on a graph to display a relative comparison between models. The graph for the accuracies across models are as follows:

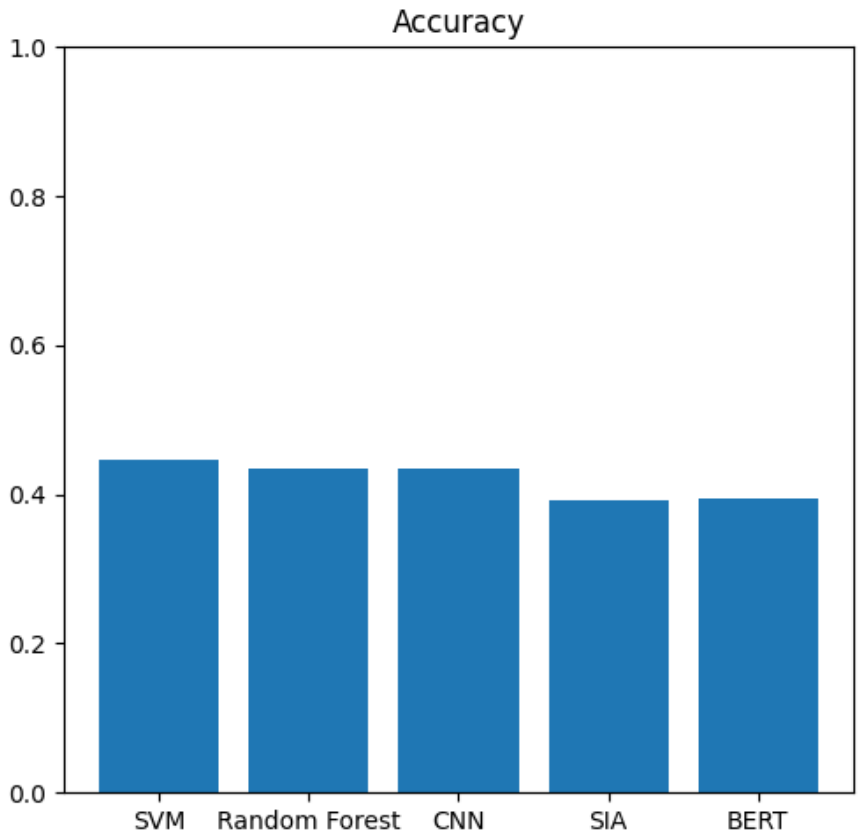

FIGURE 1. Accuracies across models

For clarification, SIA is the abbreviation for the NLTK Sentiment Analysis model we ran on the dataset. As can be seen from the graph, the accuracies for the models hover around the 40% mark implying an extremely low rate of success when predicting the movement of a stock price. While a 40% accuracy implies that there exists a relationship between the text of a financial article and its sentiment, and the movement of a stock price, it might not be significant enough to accurately predict a label most of the time.

We were also able to plot the changes in accuracy across epochs for the BERT and CNN models. The graph for changes in accuracy for the models are as follows:

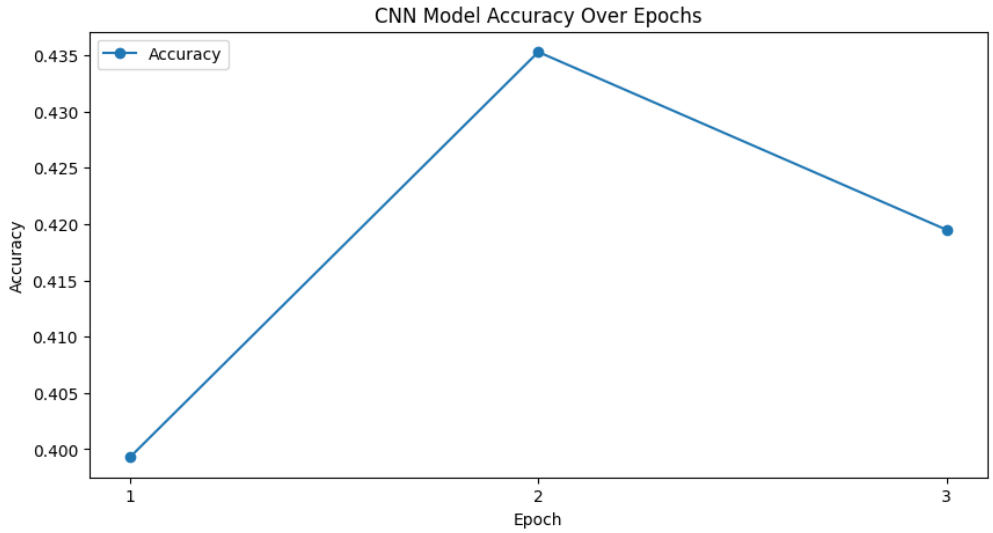

FIGURE 2. Accuracy over Epochs for CNN

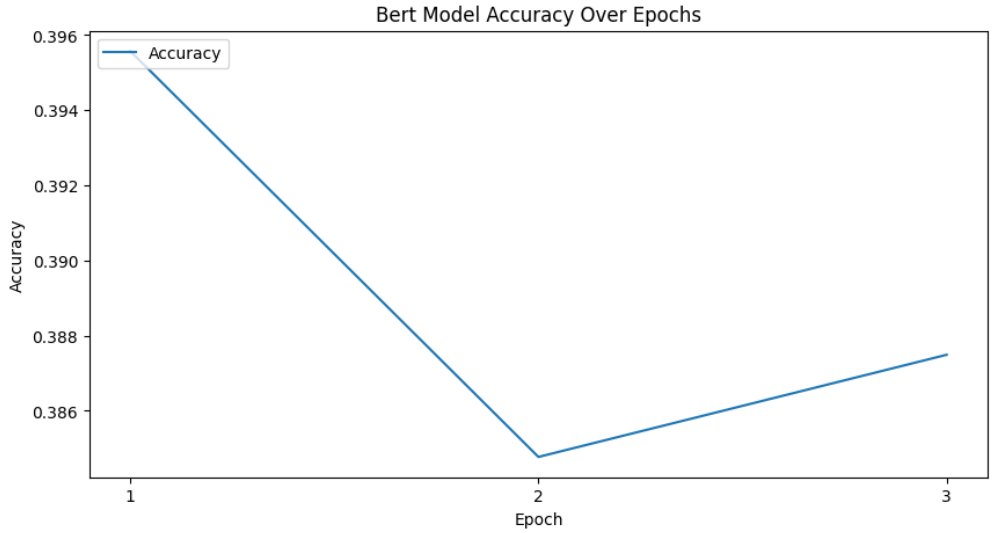

FIGURE 3. Accuracy over Epochs for BERT

For CNN, we see accuracy sharply increase from the first to the second epoch but then drop in the third epoch. This shows that the model could benefit from stopping early to prevent overfitting. In contrast, we see the accuracy drop in BERT from the first to the second epoch and then increase slightly in the third. This might imply that the model was unable to generalize the training data or that the learning rate was not optimal enough for this task.

6.2. **Precision.** The precisions for our models hover around 40% for the SVM and Random Forest models while CNN and SIA are a little lower. BERT has the lowest precision out of all the models.

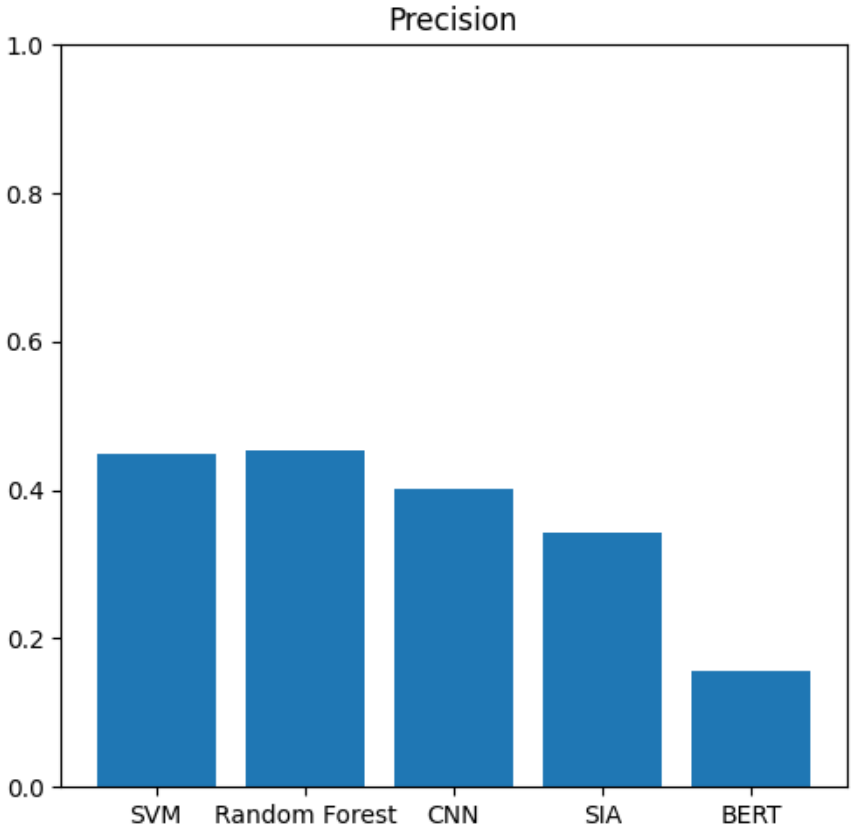

FIGURE 4. Precisions

We postulate that this might be due to the nature of the articles that were used for the purpose of this investigation. BERT's strength is a deep contextual understanding of a text and the nuances that exist within the semantic structures of the text. A lot of the articles were small to medium sized which might result in BERT not having enough information to provide an accurate analysis resulting in a high number of false positives.

6.3. **Recall.** The graphs for the recall values are as follows:

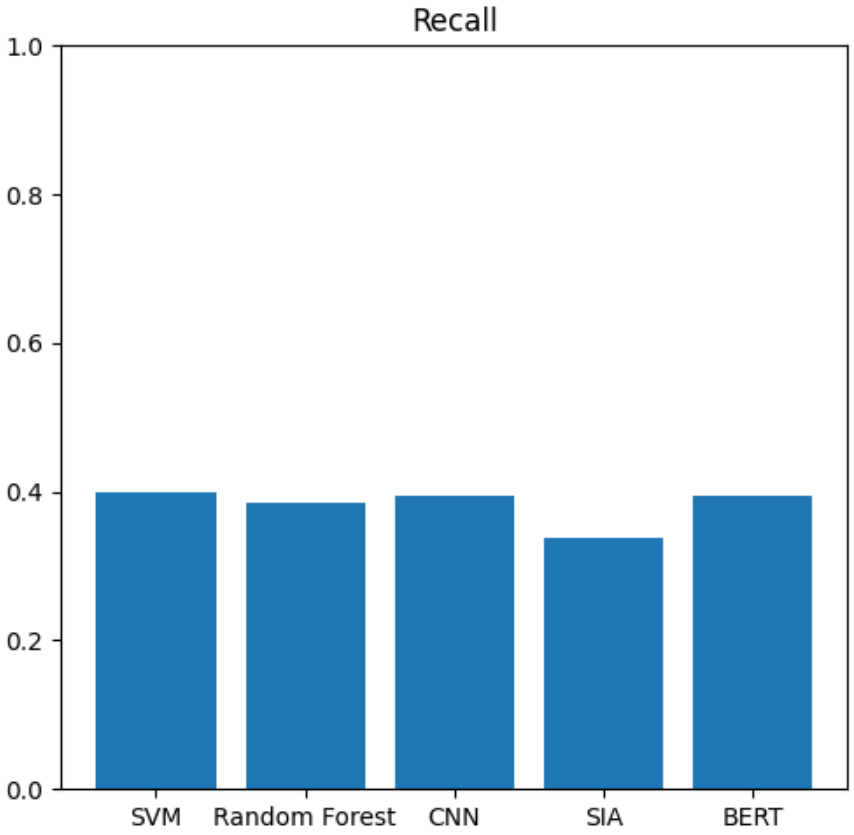

FIGURE 5. Precisions

The recall for an NLP model is an indicator of how many positives the model was able to accurately predict. A consistently lower recall is observed across all models. This could mean a low number of false positives were labeled by the models, i.e., instances where models predicted something true as false. In our case, this would mean that decreases were not predicted as increases as often (or vice versa). This would normally indicate a higher level of reliability, however when taken in conjunction with the low accuracy, this might not be true.

6.4. **F-1 Score.** The F-1 score is a metric that combines both the recall and precision into one metric to display a more balanced evaluation of a model. It is a more robust metric than a precision or recall value alone as it is the harmonic mean of them both. A high F-1 score indicates that a model might have a low number of false positives and false negatives indicating that the predictions were accurate while a low F-1 score indicates the opposite.

The lower F-1 Scores for BERT indicates that the BERT model might not be the most effective model for this task. As mentioned above, this might be due to the shorter nature of the article bodies indicating area for potential future improvements.

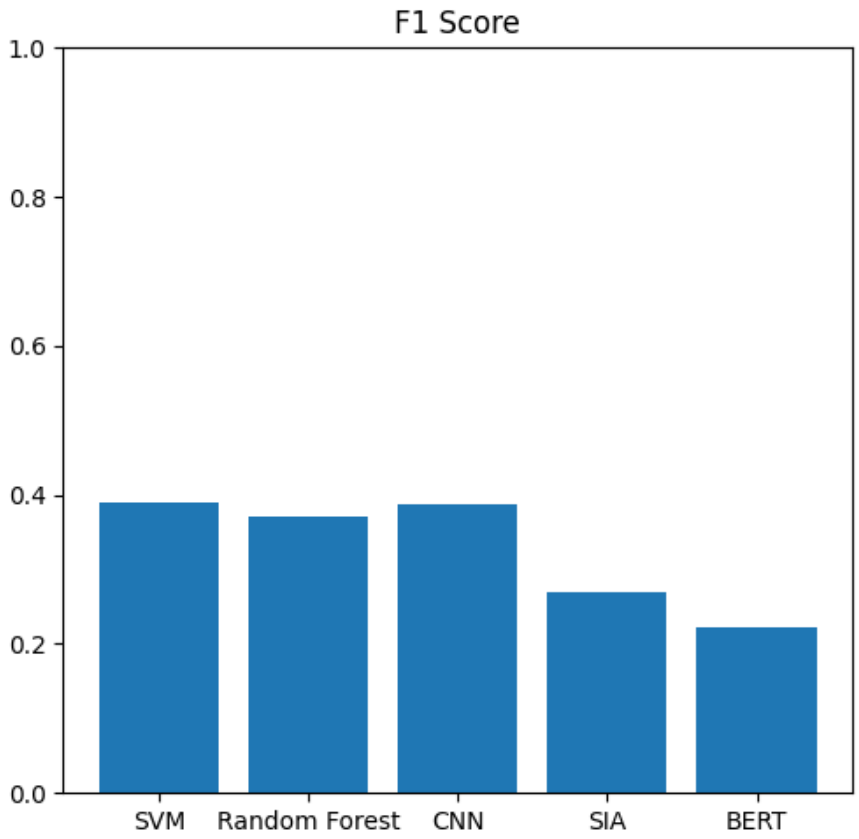

FIGURE 6. F-1 Scores

7. **Challenges**

Addressing the multitude of challenges in this project was a difficult task. The computational intensity of our model posed a significant hurdle, as we pushed the capabilities of GPU acceleration to their limits. Despite utilizing Google Colab's resources, we were frequently met with timeouts that hindered progress and underscored the need for either more advanced computational power or model optimization.

Data acquisition was a multifaceted challenge, starting with our attempt to scrape data from Benzina's website. The site's anti-scraping measures thwarted our efforts, providing us with nothing but unusable data. The situation improved when we gained access to Benzina's API. However, this solution came with its own set of constraints, primarily a request limit that acted as a bottleneck, slowing our data collection rate and forcing us to change the rate at which we made requests.

We also encountered difficulties with procuring financial data from Yahoo due to the absence of an official API. However, we were able to find a third-party service that, while less direct than an official source, offered a valuable alternative, thus allowing the project to progress.

Another issue we ran into was that our convolutional neural network (CNN) initially showed extreme overfitting, with validation accuracy reaching 99%, only to be starkly contrasted by a 20% accuracy on the test set. We managed to mitigate this by introducing early

stopping in our training process, which prevented the model from over-learning the training data and improved its generalization to new data.

The extensive and diverse nature of the data required a substantial data-cleaning effort. Achieving consistency and reliability in the datasets was particularly challenging due to their varied formats. This part of the project required careful analysis and considerable time to ensure the data's quality before it could be deemed suitable for analysis.

8. **Future Improvements**

Several strategies can be implemented. Firstly, utilizing more reputable sources like Bloomberg for training our models could be beneficial, instead of relying on sources like Benzinga, which may lack public reach.

Additionally, fine-tuning our models with various hyperparameters is essential to tailor the model specifically to our problem, potentially enhancing accuracy. Further, an emphasis on data cleaning is crucial, particularly in removing promotional content prevalent in the Benzinga API datasets used. Lastly, investing in superior GPU resources could significantly expedite data processing, enabling the handling of larger datasets more efficiently.

9. **Conclusion**

In our journey of trying to investigate the impact of long-form financial news articles on stock market fluctuations, we came to understand various characteristics of the different NLP models. Although all of our models showed consistent accuracies of around the 40%, we established that financial articles may not have the most effect on stock price changes even when using models meant to parse the complex relationships and contexts within the articles. The core of our research does not lie purely in the numerical outcome but also in a broader interpretation of these results. It displays the complexity within the various variables that impact stock market dynamics where numerous variables intertwine to influence market behavior. Our models' modest performance highlights this complexity and indicates that factors beyond the scope of financial news articles may play a more substantial role in the market. Now as we acknowledge the limitations of our studies, the constraints in our models' accuracies and our potential oversights in considering other factors provide scopes for improvement upon multiple aspects in future research. This study's outcome opens up avenues for future investigations to incorporate other variables such as social media sentiment, economic indicators, etc for enhancing our NLP models in financial contexts.

This research also tells investors and financial investors about the impact that financial articles and others may have on the role financial news articles may play in the stock market. This insight could offer a more diversified approach to analyzing market predictors while considering a wider range of sources. In conclusion, while our study sheds light on the relationship between financial articles and stock market fluctuations, it also paves the way for further exploration in this domain.

10. **Citations and References**

- Robert P. Schumaker a, et al. "A Quantitative Stock Prediction System Based on Financial News." Information Processing & Management, Pergamon, 29 May

2009,
www.sciencedirect.com/science/article/pii/S0306457309000478#aep-section-id14.

- Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter Mood Predicts the Stock Market." Journal of Computational Science, vol. 2, no. 1, 2011, pp. 1-8, doi:10.1016/j.jocs.2010.12.007.