Stock Market Indices Prediction Using Terms from Daily News Headlines

Chahat Sharma, Asst. Professor, IPEC Sahibabad, India, chahat.sharma@ipec.org.in

Abhas Omar, IPEC Sahibabad, India, abhasom.02@gmail.com

Anant Kumar Pandey, IPEC Sahibabad, India, anantpandey423@gmail.com

Ankur Kanaujia, IPEC Sahibabad, India, ankurkanaujia95@gmail.com

Devendra Kumar Rai, IPEC Sahibabad, India, devendra.rai.1910@gmail.com

Abstract **Stock market analysis is one of the biggest areas of interest for text mining. Many researchers proposed different approaches that use text information for predicting the movement of stock market indices. Many of these approaches focus either on maximizing the predictive accuracy of the model or on devising alternative methods for model evaluation. In this paper, we are going to use daily world news headlines from Reddit to predict the opening value of the Dow Jones Industrial Average. The data for this project comes from a dataset on Kaggle, and covers nearly eight years (2008–08–08 to 2016–07–01). Recently, most of NLP research have been using word embeddings to represent text, where each word or token is mapped into a numerical vector that represents the semantics of this word. Therefore, words that have similar meaning should have closer word vectors. There are several algorithms for word embedding representations. Word2Vec, GloVe, tf-idf are some of the most popular techniques. Each headline text could be represented as an aggregation or concatenation of its word vectors. For this project, we are going to use GloVe's larger common crawl vectors to create our word embeddings and Keras to build our model. We will use CNNs followed by RNNs, along with LSTMs. To help construct a better model, we will use a grid search to alter our hyperparameters' values and the architecture of our model.**

Keywords — Text Mining, Stock Market, Machine Learning, GloVe, CNN, LSTM

I. INTRODUCTION

Predicting the movement of stock market indices is of great importance to entire industries. The investors determine stock prices by using publicly available information to predict how the stock market will react, where 'publicly available information' means mostly (financial) news. Nowadays, news comes almost exclusively via web sources in the form of text. This is the reason why many researchers have proposed methods that use text information for analyzing the stock market leading to the establishment of an entirely new sub-field of data mining called text mining [1].

Recent research in predicting stock market from textual information incorporates knowledge from the fields of economy, statistics, data mining and natural language processing. There are a few main directions that the researchers tend to follow. Shynkevich et al. [2] focus on improving the predictive power of generated models by carefully choosing the modelling algorithm while simultaneously increasing and diversifying the news sources. Gidófalvi [3] concentrates on the time series nature of stock prices and uses a Naïve Bayes classifier to find the optimal 'window of influence' where the effect of news to the stock price is greatest. Fung, Yu, and Lu [4] take this idea even further by introducing complex time series segmentation methods and incorporating advanced data mining and text mining techniques in the system architecture.

Ichinose and Shimada [5] argue that 'it is unclear whether the improvement of a classifier, such as "raising" or "dropping" of a stock price of each day, contributes to the real trading.' They are doubtful whether small improvements in the classical evaluation metrics, such as prediction accuracy, recall and/or precision rate led to the improvement of an actual return in trading. They propose a trading simulation system that can estimate the improvement of an actual return in trading and experimentally show its effectiveness. They show that by using their system they can easily understand the effectiveness of one-day classifiers in terms of the real trading situation.

 DOI : 10.35291/2454-9150.2021.0460

The works of Bollen, Mao, and Zeng [6] and Chowdhury, Routh, and Chakrabarti [7] fall into the category of papers that describe the use of sentiment in text to predict the stock market. While Bollen, Mao and Zeng use sentiment analysis on tweets, Chowdhury, Routh and Chakrabarti, try to extract sentiment from news. Many other research papers have been written on the subject of 'predicting the stock market from news information' but they can all be categorized in one or more of the above-mentioned categories (predictive power improvement, time series segmentation, trading simulation evaluation and/or sentiment analysis).

Our approach, on the other hand, is to build a prediction model that will use the textual information from 'today's' Reddit top 10-12 news headlines to predict 'tomorrow's' rise or fall of the DJIA index.

II. Data Description

We used data from two independent sources:

- **News data:** Historical news headlines from Reddit World News Channel. They are ranked by Reddit users' votes, and only the top 10-12 headlines are considered for a single date.
- **Stock data:** Dow Jones Industrial Average (DJIA) daily index values were used as shown in **Figure 2.1**. On each date, the 'open,' 'high,' 'low,' 'close' and 'volume' values are recorded.

Data for the past eight years was collected – from 1 June 2013 to 30 June 2021.

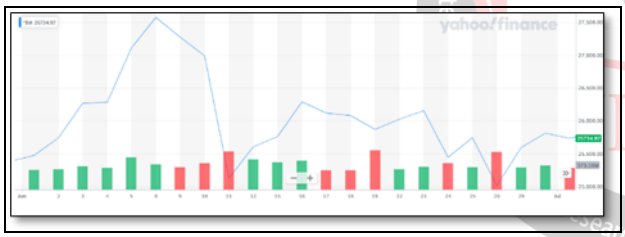

Figure 2.1 - DJIA Index (1st June 2021 to 30th June 2021)

Figure 2.1 shows the 'open' values over the June month (of the collected data).

We divided the work in four phases: Data transformation, Modelling technique selection, Quantitative evaluation and Qualitative evaluation.

The images as PostScript (PS), Encapsulated PostScript (EPS), or Tagged Image File Format (TIFF), sizes them, and adjusts the resolution settings. If you created your source files in one of the following you will be able to submit the graphics without converting to a PS, EPS, or TIFF file: Microsoft Word, Microsoft PowerPoint, Microsoft Excel, or Portable Document Format (PDF).

III. Data Transformation Phase

The data for this project is in two different files. Due to this, we need to ensure that we have the same dates in each of our data frames. The function isin() helped us here.

To create our target values, we took the difference in opening prices between the current and following day. Using this value, we were able to see how well the news will be able to predict the change in opening price.

```
dj = dj.set_index('Date').diff(periods=1)
dj['Date'] = dj.index
dj = dj1.reset_index(drop=True)
```

Now that we have our target values, we needed to create a list for the headlines in our news and their corresponding price change.

Each day, for the most part, included 10-12 headlines. This is what made up our 'news' data. We needed to clean this data to get the most signal out of it. To do this, we converted it to the lower case, replaced contractions with their longer forms, removed unwanted characters, reformatted words to better match GloVe's word vectors, and removed stop words. "clean_text()" was the function that we have used in the algorithm in order to clean the data.

To create the weights that will be used for the model's embeddings, we created a matrix consisting of the embeddings relating to the words in our vocabulary. If a word is found in GloVe's vocabulary, we will use its pre-trained vector. If a word is not found in Glove's vocabulary, we will create a random embedding for it. The embeddings will be updated as the model trains, so our new 'random' embeddings will be more accurate by the end of training.

The final step in preparing our headline data was to make each day's news the same length. We maximized the length of any headline to 16 words (this is the length of the 75th percentile headline) and maximized the length of any day's news to 200 words. These values were picked to have a good balance between the number of words in a headline and the number of headlines to use.

IV. Methodology

A. Convolutional Neural Network

A more capable and advanced variation of classic artificial neural networks, a Convolutional Neural Network (CNN) is built to handle a greater amount of complexity around pre-processing, and computation of data. CNNs were designed for image data and might be the most efficient and flexible model for image classification problems. Although CNNs were not particularly built to work with non-image data, they can achieve stunning results with non-image data as well. After we have imported our input data into the model, there are 4 parts to building the CNN:

1. Convolution: a process in which feature maps are created out of our input data. A function is then applied to filter maps.

 DOI : 10.35291/2454-9150.2021.0460

2. Max-Pooling: enables our CNN to detect an image when presented with modification.

3. Flattening: Flatten the data into an array so CNN can read it.

4. Full Connection: The hidden layer, which also calculates the loss function for our model.

A. LSTM

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN (**Figure 4.1)**, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs **(Figure 4.2)** are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

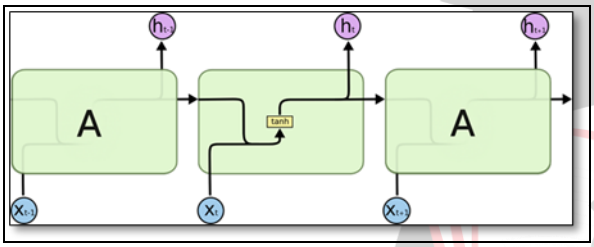

Figure 4.1 Recurrent Neural Network (RNN) Model

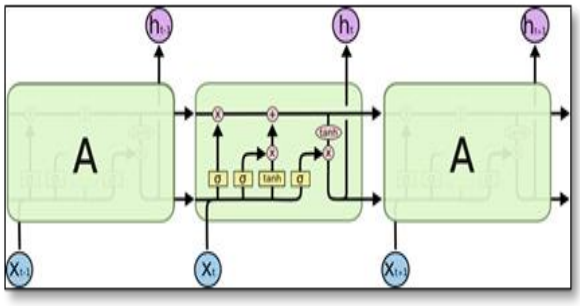

Figure 4.2 Long Short-Term Memory (LSTM) Layers

V. IMPLEMENTATION OF MODELS/ALGORITHM

In simple words the algorithm can be described by the following points:

- We will collect the stock data
- Pre-process the data i.e., Train and Test
- Create a stacked CNN and LSTM model
- Predict the sample data and plot the output

Convolutional neural networks (CNNs) are likely to extract local and deep features from natural language. It has been shown that CNN has gotten improved results in sentence classification. Recurrent neural networks (RNNs) are various kinds of time-recursive neural network that is able to learn the long-term dependencies in sequential data. Seeing that we can view the words in a sentence as a sequence from left to right, RNNs can be modeled in accordance with people's reading and understanding behavior of a sentence. Socher et. al [8] presented a convolutional-recursive deep model for 3D object classification that combined the convolutional and recursive neural networks together. The CNN layer learns the low-level translation invariant features which are inputs to multiple, fixed-tree RNNs (recursive neural networks) in order to compose higher order features. Kim [9] described a model that employed a convolutional neural network (CNN) and a highway network over characters, whose output is given to a long short-term memory (LSTM) recurrent neural network language model (RNN-LM). These two models both get better results than prior methods. However, the recursive neural networks need to build a tree structure that is usually based on the parser result of sentence. The recurrent neural network is particularly suited for modeling the sequential pattern. Inspired by those works and the fact that CNN can extract local features of input and RNN (recurrent neural network) can process sequence input and learn the long-term dependencies, we combine both of them in the analysis of short texts. Our model consists of the following parts: word embeddings and sentence-level representation, convolutional and pooling layers, concatenation layer, RNN layer, LSTM.

VI. CONCLUSION

The research focuses on either predictive power improvement, time series segmentation, trading simulation evaluation, sentiment analysis or a combination of those. Our approach, on the other hand, tries to exploit the descriptive power of the predictive models by analyzing the (combination of) words that they contain and associate these to the movement of the stock price. By binarizing the stock price movement, we used the machine learning approach to learn models that are able to predict the rise or fall of the stock.

This dataset is a short dataset of news, including only 10-12 news.

One major outcome for us was important sampling to get better predictions. We were initially getting validation accuracy as low as 20 % and when we investigated the issue, we realized NumPy random sampling is not able to shuffle the dataset properly. Our training dataset had for example categories 1, 2, 3 and we were trying to validate data with categories 3 and 4. We implemented stratified sampling in order to solve this problem. This improved our validation accuracy by three times.

The main contribution of this work is showing that by using appropriate machine learning algorithms one can accurately predict individual words which almost always (or

 DOI : 10.35291/2454-9150.2021.0460

practically never) co-occur with the movement of a selected stock market index (DJIA in our case).

We regard this as a preliminary study of descriptive analysis of predictive models for stock markets using textual data. A first step has been done in this direction by showing that there is some relation between the words in the headlines of the daily news and the movement of the stock market price. This research concentrates on classification methods for building prediction models, which can predict only the sign of the stock movement (rise or fall). More formal ways should be studied to prove this relation and quantify its extent.

REFERENCES

[1] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[2] Y. Shynkevich, T.M. Mcginnity, S. Coleman and A. Belatreche, "Forecasting movements of Health-Care stock prices based on different categories of news articles using multiple kernel learning", Decision Support Systems, 2016

[3] G. Gidófalvi, "Using News Articles to Predict Stock Price Movements", 2001

[4] http://cseweb.ucsd.edu/~elkan/254spring01/gidofalvirep.pdf

[5] K. Ichinose and K. Shimada, "Stock Market Prediction from News on the Web and a New Evaluation Approach in Trading", 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2016

[6] J. Bollen, H. Mao and X. Zeng "Twitter Mood Predicts the Stock Market", Journal of Computational Science 2(1), 2010

[7] S. Chowdhury, Soham Routh and Satyajit Chakrabarti "News Analytics and Sentiment Analysis to Predict Stock Price Trends", IJCSIT, 2014

[8] R. Socher, B. Huval, B. Bhat, C. D. Manning and A. Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification", 2012

[9] Y. Kim, Y. Jernite and D. Sontag and A. M. Rush, "Character-Aware Neural Language Models", Computation and Language

[10] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IJREAM Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

 DOI : 10.35291/2454-9150.2021.0460