Deep Learning for Stock Movement Prediction

Group Members

Name: Lee Wasin	**Student ID:** 20266284	**Git:** Wasssssss
Name: Kung Wai Tat	**Student ID:** 20240527	**Git:** peterkung543@gmail.com
Name: Chan Nok Hin	**Student ID:** 20349103	**Git:** samsonchan666@gmail.com

1. Abstract

This project aims to predict stock movement by sentiment analysis of stock news, specifically Apple Inc. (AAPL). We scraped stock price and news, labeled the news “positive” or “negative” according to the stock movement after a specific date interval, and then trained a prediction model using deep learning method. With the help of AWS EC2 and Spark, a large amount of data can be scraped, pre-processed and trained easily. The result shows that sentiment of stock news is highly correlated to short-term stock movement, and a 62% of accuracy is achieved for predicting stock movement.

2. Introduction

Stock has been one of the major investing tools for many years. According to the efficient market hypothesis, in an ideal market, asset prices should fully reflect all available information, meaning stock movement is based on investors’ trading strategies and these strategies are normally made based on the information they gathered. In recent years, with the improved technologies and communication systems, investors had all the latest financial news at their fingertips, which has resulted in changing their investment decisions in a very short of time. As a result, the market fluctuates quickly, making it harder to perform prediction. Yet with the development in machine learning, stock prediction is no longer an impossible task.

In this project, we decided to predict the stock movement of AAPL by analyzing the sentiment of news articles, and for the algorithm, we use Convolutional Neural Network (CNN) as a deep learning algorithm.

3. Develop Environment

As the system requires high computation power, we launched an EC2 P3 instance, which computed in the cloud with up to 8 NVIDIA Tesla V100 GPUs to increase speed in training.

4. Data

4.1. Data Scraping

Python scrapers were implemented to scrape stock prices and news for approximately past one year.

The close prices of AAPL was scraped from Yahoo finance API and the “Date” and “Close” columns were stored. The news related to AAPL was scraped from 6 websites which included “Thestreet”, “Zacks”, “InvestorPlace”, “Investor's Business Daily”, “Gurufocus” and “The Motley Fool” from 1 June 2017 to 1 July 2018. The date and news content were stored. A total of 7746 news was scraped.

Both stock prices and news were saved in form of Python pickle for easy access.

4.2. Data Cleaning

4.2.1. Stock prices

We assume the sentiment of a news will affect the corresponding stock movement n days after the release of the news (with n going to be found in the result part). However, there were some missing stock prices on public holidays or weekend. Therefore, a simple equation is used to fill up the missing prices:

$$P_{fill} = \frac{P_{prev} + P_{next}}{2}$$

P_{fill} is the missing price, P_{prev} is the price of previous date, P_{next} is the price the date which contains the next price. For example, if both Saturday and Sunday are missing, the price on Saturday will be the average of Friday and Monday.

4.2.2. Stock news

The sentiment words in news play a crucial role in opinion mining. Thus, news contents were cleaned in preparation for word dictionary building and deep learning. The diagram below shows the flow of stock news cleaning.

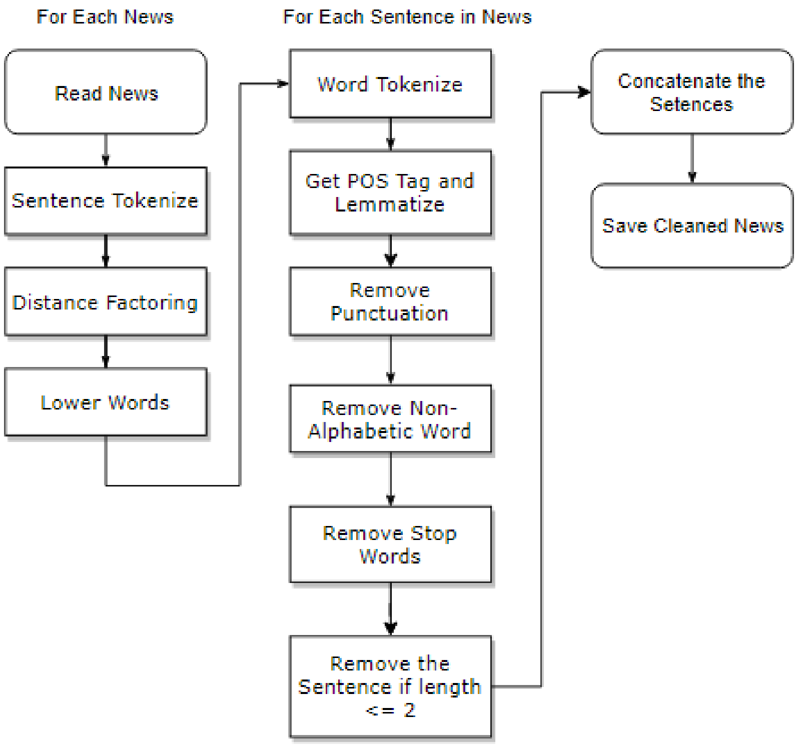


Figure 1: Flow of text cleaning

Distance factoring was a technique implemented by [1]. One of the problems we encountered was that not every sentence in an article was about AAPL stock. Thus, some sentences might not be relevant, which consequently affected the performance of training. For example, competitors were always mentioned in a news for a company, so words like “income growth” may be referenced to competitors. Therefore, when the target company “AAPL” was mentioned in a sentence, only the current, preceding and succeeding sentences were taken while others should be ignored.

Lemmatization was performed to ensure different forms of a word were lemmatized to the same word. The list of stop words were provided by NLTK library. Some words were removed as they were related to stock movement, which included

[1] Alex Brojba-Micu, StockWatcher 2.0: Using Text Analysis to Predict Stock Market Trends, 2013.

"up", "down", "below", "won", "further", "most", "few", "after", "against" and "own". If the length of the sentence after these removals is smaller than 3, we considered the sentence was not useful and ignored it. Last, relevant sentences were concatenated and stored.

4.2.3. Dictionary Building

As mentioned by [2], to predict stock price movements, a stock domain specific dictionary showed greater accuracy when compared with general sentiment dictionaries. Therefore, this dictionary was built following the steps below.

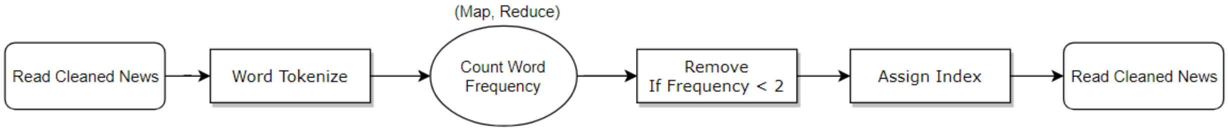
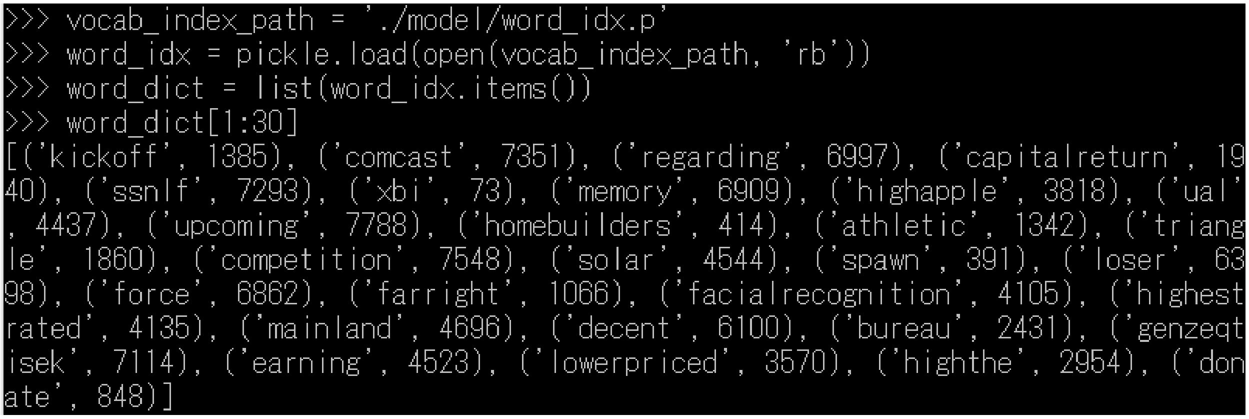


Figure 2: Flow of dictionary building

As a total of 411845 words were needed to be handled, Spark was used word frequency counting, removal and indexing. Words that had frequency less than 2 were removed as they were not useful. This greatly reduced the size of the dictionary. The dictionary was then saved as Python pickle. The figure below shows some samples in our dictionary.



```
>>> vocab_index_path = './model/word_idx.p'
>>> word_idx = pickle.load(open(vocab_index_path, 'rb'))
>>> word_dict = list(word_idx.items())
>>> word_dict[1:30]
[('kickoff', 1385), ('comcast', 7351), ('regarding', 6997), ('capitalreturn', 19
40), ('ssnlf', 7293), ('xbi', 73), ('memory', 6909), ('highapple', 3818), ('ual'
, 4437), ('upcoming', 7788), ('homebuilders', 414), ('athletic', 1342), ('triang
le', 1860), ('competition', 7548), ('solar', 4544), ('spawn', 391), ('loser', 63
98), ('force', 6862), ('farright', 1066), ('facialrecognition', 4105), ('highest
rated', 4135), ('mainland', 4696), ('decent', 6100), ('bureau', 2431), ('genzeqt
isek', 7114), ('earning', 4523), ('lowerpriced', 3570), ('highthe', 2954), ('don
ate', 848)]
```

Figure 3: Samples in word-counting dictionary

5. Training

5.1. *Training set*

The cleaned articles were labeled according to the stock movement n days (hyperparameter to be examined) after the release of the news, 1 for stock rise, 0 for stock fall. The data was shuffled randomly, with 90% split for training and 10% for testing.

5.2. *Neural Network*

Keras was used as the deep learning API to construct the network due to its convenience for prototyping and support for CNN. The diagram in the next page shows the architecture of our model.

[2] Yoosin Kim, Seung Ryul Jeong, Imran Ghani, Text Opinion Mining to Analyze News for Stock Market Prediction, Int. J. Advance. Soft Comput. Appl., Vol. 6, No. 1, 2014.

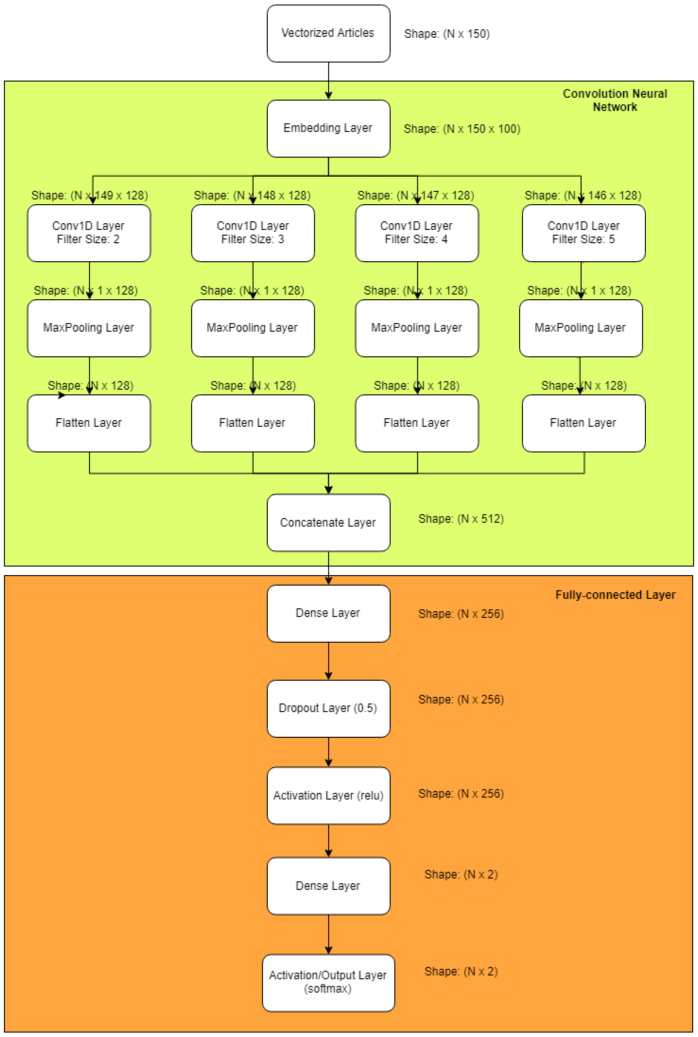


Figure 4: Architecture of CNN model for stock news

The cleaned articles were vectorized by assigning an index for each word using the dictionary built. We padded the sequence to 150 length as almost all articles were within this length after cleaning.

The vectorized articles continued to pass through an embedding layer with output dimension of 100. Then, it passed through diverse 1-d convolution layers with different kernel sizes. We chose to use kernel size 2, 3, 4, 5, which referred to bi-gram, tri-gram and so on. The number of output filters in convolution was 128. After that, they were max-pooled, flattened and concatenated.

Lastly, it passed through a fully-connected layer and yielded binary output - the probability that the news article was positive or negative.

In the training, Adam optimizer was used with learning rate 0.0001.

6. Results

We tried different time lag between the release date of a news and stock movement. It was found that time lag n = 4 days yielded the most accurate result.

The model obtained the best accuracy of 62% for correct classification of news sentiment. The graphs below showed the validation loss and accuracy during training.

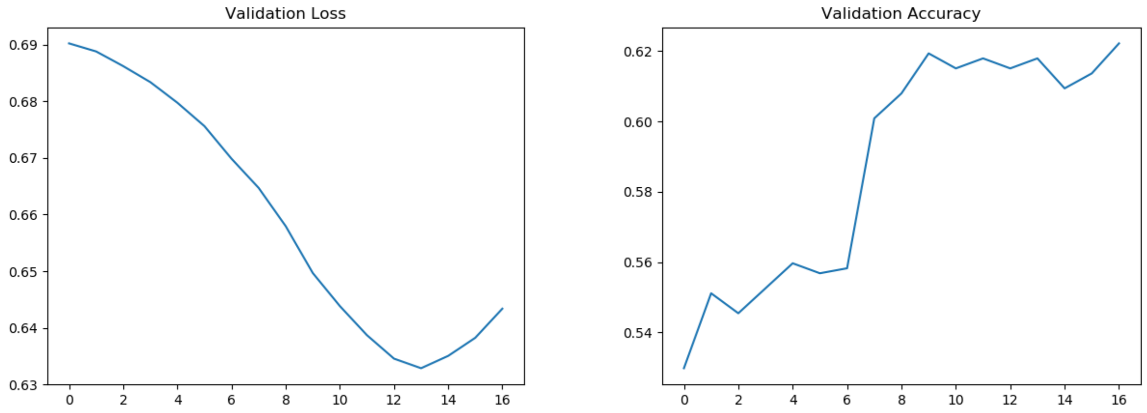


Figure 5: Validation loss and accuracy for different times of training

Statistical summary:

	precision	recall	f1-score	support
0	0.62	0.51	0.56	331
1	0.62	0.72	0.67	373
avg / total	0.62	0.62	0.62	704

Accuracy: 0.6221590909090909

The model made prediction per news. We grouped the news by day and compared it with the stock movement. The sentiment score was the average score of articles by date.

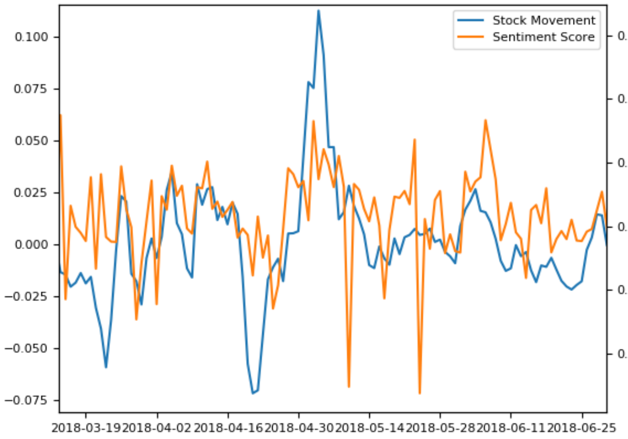


Figure 6: Stock movement vs Sentiment score for AAPL

7. Conclusion

In this project, we predicted the stock movement of AAPL using stock news. We implemented (1) Scraper for stock price and news; and (2) Deep learning model for training. The result shows a strong relationship between news sentiment and stock movement. We achieved 62% accuracy for correct classification of the news sentiment.

Yet, as this project only focus on how words or phrases correlated to stock movement, further work can be extended to analyze the semantic meaning of these words.

Source code: https://github.com/samsonchan666/comp4651-Project