

Final Year Project

SCSE 19-0125

AI Based Stock Market Trending Analysis

Interim Report

Student Name: Chan Wen Jun, Lawrann

Matriculation Number: U1721447K

Supervisors: Dr Wang Zhaoxia, Dr Li Fang

**Table of Contents**

[**Progress on Data Collection** 3](#_Toc30882511)

[Twitter Data Extraction 3](#_Toc30882512)

[New York Times Data Extraction 6](#_Toc30882513)

[Stock Data Extraction 6](#_Toc30882514)

[Issues Encountered 7](#_Toc30882515)

[**Progress on Data Cleaning / Preprocessing** 8](#_Toc30882516)

[Cleaning Twitter Dataset 8](#_Toc30882517)

[Cleaning New York Times Dataset 9](#_Toc30882518)

[Issues Encountered 10](#_Toc30882519)

[**Progress on Natural Language Processing** 11](#_Toc30882520)

[BERT 11](#_Toc30882521)

[Sentiment Analysis on Twitter Tweets 12](#_Toc30882522)

[Sentiment Analysis on News Articles 14](#_Toc30882523)

[Issues Encountered 15](#_Toc30882524)

[**Progress on LSTM Models** 16](#_Toc30882525)

[**Identifying Improvements** 16](#_Toc30882526)

[Further Data Cleaning for News Articles 16](#_Toc30882527)

[Further Fine Tuning 16](#_Toc30882528)

[Further Tuning of Hyper Parameters 16](#_Toc30882529)

# **Progress on Data Collection**

## Twitter Data Extraction

Twitter tweets have been extracted for Citigroup Inc. and S&P 500 Index.

Initially, I tried using Twitter REST API to pull historical tweets relating to the respective companies. However, there is a rate limit implemented on Twitter of 180 request every 15 minutes with maximum of 100 tweets per request. This limited the number of tweets I can extract to 72,000 tweets an hour. Additionally, the search API of twitter only allows for access of tweets within the past seven days meaning I would be unable to pull historical tweets dating back to 2014.

To overcome this issue, I found a MIT Licensed Twitter scraper on Github by user taspinar (<https://github.com/taspinar/twitterscraper/>). This Twitter scraper allows the use of proxies to bypass the rate limit and also allowed me scrape historical tweet past the seven-day search limit of Twitter API.

Using the Twitter scraper and some proxies I obtained, I scrapped tweets from 1st January 2014 to 31st December 2018 relating to Citigroup Inc. and S&P 500 Index.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| citigroup tweets | | | | | | | | | | |
| 2014 | | **2015** | | **2016** | | **2017** | | **2018** | | |
| Month | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** |
| 1 | 11,452 | **1** | 16,813 | **1** | 15,285 | **1** | 14,034 | **1** | 10,415 |
| 2 | 9,413 | **2** | 16,586 | **2** | 16,138 | **2** | 8,469 | **2** | 10,756 |
| 3 | 19,288 | **3** | 25,225 | **3** | 13,215 | **3** | 7,343 | **3** | 15,654 |
| 4 | 21,749 | **4** | 15,376 | **4** | 15,979 | **4** | 8,946 | **4** | 11,084 |
| 5 | 14,336 | **5** | 15,743 | **5** | 11,506 | **5** | 9,067 | **5** | 9,841 |
| 6 | 15,113 | **6** | 14,880 | **6** | 14,031 | **6** | 7,675 | **6** | 12,258 |
| 7 | 33,203 | **7** | 19,329 | **7** | 15,250 | **7** | 11,537 | **7** | 9,636 |
| 8 | 11,728 | **8** | 17,025 | **8** | 16,580 | **8** | 6,994 | **8** | 10,409 |
| 9 | 10,485 | **9** | 13,129 | **9** | 10,509 | **9** | 8,221 | **9** | 11,010 |
| 10 | 17,666 | **10** | 15,690 | **10** | 13,046 | **10** | 12,055 | **10** | 9,485 |
| 11 | 13,233 | **11** | 9,915 | **11** | 10,172 | **11** | 10,584 | **11** | 9,278 |
| 12 | 24,946 | **12** | 11,899 | **12** | 7,674 | **12** | 8,333 | **12** | 6,691 |
| Total | 202,612 | **Total** | 191,610 | **Total** | 159,385 | **Total** | 113,258 | **Total** | 126,517 |

Search parameters used for obtaining Citigroup related tweets:

* #Citi
* NYSE:C
* $C
* Citigroup

A total of **793,382** tweets was obtained from 1st January 2014 to 31st December 2018

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S&P 500 TWEETS | | | | | | | | | | |
| 2014 | | **2015** | | **2016** | | **2017** | | **2018** | | |
| Month | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** |
| 1 | 24,622 | **1** | 17,035 | **1** | 20,878 | **1** | 22,492 | **1** | 24,836 |
| 2 | 30,088 | **2** | 11,689 | **2** | 15,288 | **2** | 20,773 | **2** | 30,233 |
| 3 | 28,708 | **3** | 14,184 | **3** | 21,182 | **3** | 24,022 | **3** | 28,881 |
| 4 | 29,514 | **4** | 13,627 | **4** | 17,529 | **4** | 19,826 | **4** | 23,218 |
| 5 | 32,241 | **5** | 22,543 | **5** | 24,932 | **5** | 27,329 | **5** | 17,363 |
| 6 | 32,464 | **6** | 11,104 | **6** | 22,211 | **6** | 21,244 | **6** | 19,915 |
| 7 | 24,125 | **7** | 10,977 | **7** | 37,108 | **7** | 22,126 | **7** | 19,490 |
| 8 | 40,253 | **8** | 24,301 | **8** | 25,422 | **8** | 27,067 | **8** | 26,566 |
| 9 | 26,487 | **9** | 25,328 | **9** | 23,180 | **9** | 24,277 | **9** | 20,563 |
| 10 | 28,880 | **10** | 20,692 | **10** | 20,247 | **10** | 25,066 | **10** | 29,544 |
| 11 | 26,426 | **11** | 15,899 | **11** | 25,738 | **11** | 22,673 | **11** | 23,485 |
| 12 | 24,882 | **12** | 23,648 | **12** | 20,427 | **12** | 22,044 | **12** | 36,464 |
| Total | 348,690 | **Total** | 211,027 | **Total** | 274,142 | **Total** | 278,939 | **Total** | 300,558 |

Search parameters used for obtaining S&P 500 Index related tweets:

* S&P 500
* S&P500
* INDEXSP
* #S&P500

A total of **1,413,356** tweets was obtained from 1st January 2014 to 31st December 2018

The meta data scrapped for each tweet includes the fullname, html, is\_retweet, likes, replies, retweet\_id, retweeter\_userid, retweeter\_username, retweets, text, timestamp, timestamp\_epochs, tweet\_id, tweet\_url, user\_id, username.

The following table shows the description of tweets obtained using the Twitter scrapper.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Example** |
| fullname | The display name of the user. | Efendi Achmad |
| html | The html of the tweet. | <p class=\"TweetTextSize js-tweet-text tweet-text\" data-aria-label-part=\"0\" lang=\"en\">US Equities finished mixed on the day \u2013 DJIA: +41.55pts and <strong>S&amp;P500</strong>: -7.19pts ^CT <a class=\"twitter-hashtag pretty-link js-nav\" data-query-source=\"hashtag\_click\" dir=\"ltr\" href=\"/hashtag/stocks?src=hash\"><s>#</s><b>stocks</b></a></p> |
| is\_retweet | States whether the tweet was retweeted from an original tweet. | 0 |
| likes | The number of likes obtained from the public. | 0 |
| replies | The number of replies to the tweet. | 0 |
| retweet\_id | The unique id of the retweet. | 424688987421155329 |
| retweeter\_userid | The user id of the person who retweet the tweet. | 41280137 |
| retweeter\_username | The username of the person who retweet the tweet. | bobo3000 |
| retweets | The number of retweets obtained from the public. | 0 |
| text | The content of the tweet. | US Equities finished mixed on the day \u2013 DJIA: +41.55pts and S&P500: -7.19pts ^CT #stocks |
| timestamp | The time and date the tweet was posted. | 2014-01-18T23:49:07 |
| timestamp\_epochs | The time and date the tweet was posted but in the epochs format. | 1390088947 |
| tweet\_id | The unique id of the tweet. | 424689944242565120 |
| tweet\_url | The hyperlink paramaters to the tweet through http://www.twitter.com. | /AlodyaBirdFarm/status/424689944242565120 |
| user\_id | The unique id of the user. | 1125472027 |
| username | The login name of the user. | AlodyaBirdFarm |

## New York Times Data Extraction

I was able to use the New York Times API for extracting the relevant articles of Citigroup Inc and S&P 500 Index.

The rate limit of New York Times API is 4,000 request a day and 10 request a minute. Using the python library nytimesarticle to access the New York Time API, I queried for the keywords “S&P” and “Citi” to obtain the relevant articles of Citigroup and S&P500. Using an interval of six seconds before each query to ensure I do not hit the rate limit.

In total I obtained:

* **24,516** articles relating the query “Citi”
* **3,865** articles relating to the query “S&P”

The meta data scrapped for each article includes the id, abstract, headline, desk, date, section, snippet, lead\_paragraph, source, type, url, word\_count, location and subjects

The following table shows the description of articles obtained.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Description** | **Example** |
| id | The unique id of the article | nyt://article/c66e6348-c6f2-554b-a202-98a7dc6ff5a0 |
| abstract | The abstract paragraph of the article | Representative Nick Rahall, siding with Republicans more often than ever, is a prime example of how red-state Democrats are trying to survive. |
| headline | The headline of the article | A West Virginia Democrat Battles Extinction |
| desk | Which desk the article came from (e.g Editorial, Foreign) | Upshot |
| date | Date the article was posted | 5/1/2014 |
| section | Which section the article belongs to | The Upshot |
| snippet | A snippet of the article | Representative Nick Rahall, siding with Republicans more often than ever, is a prime example of how red-state Democrats are trying to survive. |
| lead\_paragraph | The lead paragraph of the article | It was a vote intended to split along party lines; instead, it revealed just how worried some House Democrats are about re-election. |
| source | Which source the article was from | The New York Times |
| type | The type of the article (e.g News, Letter) | News |
| url | The article’s hyperlink | https://www.nytimes.com/2014/05/02/upshot/a-west-virginia-democrat-battles-extinction.html |
| word\_count | Words within the article | 679 |
| location | The location relating to the article | ['West Virginia'] |
| subjects | The key subjects of the article | ['Elections, House of Representatives', 'United States Politics and Government'] |

## Stock Data Extraction

To obtain the relevant historical stock price data, I made use of Yahoo Finance. All historical stock data from 1st January 2014 to 31st December 2018 were downloaded from Yahoo Finance into csv files. They include the Stock’s Date, Opening, Closing, High, Low, Adjusted Closing Prices as well as Volume of Trades

## Issues Encountered

Took time to identify he proper search query which would constitute all related tweets of the relevant stock. Similarly for the news articles.

# **Progress on Data Cleaning / Preprocessing**

## Cleaning Twitter Dataset

Cleaning of Twitter tweets is done in three steps

1. Removing unwanted meta data / field name
2. Removing links <http://bit.ly/example> , hashtags #, mentions @
3. Removing duplicated tweets

Firstly, I remove unwanted meta data and only keep the username, date along with the tweet text. Secondly, I proceed to remove any links within the text, hashtags # and mentions @ and identify spam/bot account tweets for removal. Lastly, any duplicate tweets within the same day are identified and remove. Additionally, after each process of cleaning, I manually scan through the datasets for any anomalies with the cleaning and or identifying unwanted texts/patterns which can be removed.

Three python scripts were created to perform the process of cleaning.

1. tweets\_cleaner\_remove\_links\_terms.py
2. tweets\_cleaner\_remove\_unwanted\_headers.py
3. tweets\_cleaner\_remove\_duplicate.py

After preprocessing the total number of tweets for both S&P 500 and Citigroup shrunk from **2,206,738** to **1,262,667** records.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CITIGROUP TWEETS CLEANED | | | | | | | | | | |
| 2014 | | **2015** | | **2016** | | **2017** | | **2018** | | |
| Month | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** |
| 1 | 7,345 | **1** | 8,552 | **1** | 7,510 | **1** | 6,672 | **1** | 6,857 |
| 2 | 5,378 | **2** | 8,456 | **2** | 8,971 | **2** | 4,310 | **2** | 7,351 |
| 3 | 9,516 | **3** | 11,262 | **3** | 7,098 | **3** | 4,499 | **3** | 10,388 |
| 4 | 10,150 | **4** | 8,627 | **4** | 8,164 | **4** | 5,421 | **4** | 8,197 |
| 5 | 6,949 | **5** | 9,555 | **5** | 5,796 | **5** | 5,166 | **5** | 7,841 |
| 6 | 7,101 | **6** | 8,886 | **6** | 6,916 | **6** | 4,858 | **6** | 9,236 |
| 7 | 13,012 | **7** | 10,350 | **7** | 7,617 | **7** | 6,609 | **7** | 7,016 |
| 8 | 6,234 | **8** | 8,552 | **8** | 8,576 | **8** | 4,587 | **8** | 7,809 |
| 9 | 5,574 | **9** | 8,166 | **9** | 5,835 | **9** | 5,388 | **9** | 7,401 |
| 10 | 8,378 | **10** | 8,783 | **10** | 6,967 | **10** | 7,823 | **10** | 7,127 |
| 11 | 7,182 | **11** | 6,226 | **11** | 6,001 | **11** | 6,194 | **11** | 7,088 |
| 12 | 12,553 | **12** | 7,122 | **12** | 4,647 | **12** | 5,197 | **12** | 5,297 |
| Total | 99,372 | **Total** | 10,4537 | **Total** | 84,098 | **Total** | 66,724 | **Total** | 91,608 |

Citigroup total number of Tweets remaining after preprocessing: **446,339**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S&P 500 TWEETS CLEANED | | | | | | | | | | |
| 2014 | | **2015** | | **2016** | | **2017** | | **2018** | | |
| Month | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** | **Month** | **#** |
| 1 | 14,601 | **1** | 8,654 | **1** | 11,855 | **1** | 14,152 | **1** | 16,904 |
| 2 | 16,004 | **2** | 5,980 | **2** | 9,596 | **2** | 12,742 | **2** | 19,193 |
| 3 | 15,955 | **3** | 7,723 | **3** | 13,042 | **3** | 14,490 | **3** | 16,756 |
| 4 | 15,930 | **4** | 7,385 | **4** | 10,658 | **4** | 12,710 | **4** | 13,889 |
| 5 | 14,561 | **5** | 11,770 | **5** | 14,545 | **5** | 16,393 | **5** | 12,855 |
| 6 | 15,252 | **6** | 4,899 | **6** | 12,836 | **6** | 14,312 | **6** | 14,409 |
| 7 | 12,289 | **7** | 5,557 | **7** | 18,364 | **7** | 14,663 | **7** | 13,775 |
| 8 | 16,414 | **8** | 12,318 | **8** | 13,750 | **8** | 17,021 | **8** | 17,587 |
| 9 | 12,174 | **9** | 12,267 | **9** | 13,730 | **9** | 15,080 | **9** | 14,265 |
| 10 | 14,657 | **10** | 10,753 | **10** | 11,957 | **10** | 16,440 | **10** | 21,064 |
| 11 | 11,288 | **11** | 8,932 | **11** | 15,151 | **11** | 15,072 | **11** | 16,334 |
| 12 | 10,966 | **12** | 12,882 | **12** | 12,803 | **12** | 14,712 | **12** | 24,012 |
| Total | 170,091 | **Total** | 109,120 | **Total** | 158,287 | **Total** | 177,787 | **Total** | 201,043 |

S&P 500 total number of Tweets remaining after preprocessing: **816,328**

## Cleaning New York Times Dataset

Cleaning of New York Times dataset differs for both Citigroup and S&P 500. During the data extraction of articles for both Citigroup and S&P 500, I wanted to extract as much articles as possible. Hence, I used the search term “Citi” and “S&P” to get as much article coverage as possible. During the cleaning of the dataset, my aim was to remove unrelated articles.

Cleaning of New York Times articles is done in two steps

1. Filter for specific terms within the abstract, headline, lead\_paragraph or snippet field.
2. Filter for related sections

For Citigroup related articles, I removed articles that did not contain the terms “Citibank”, “Citigroup”, “Citi “, “Citi’s ”, “Citi group”, “Citi bank” within the abstract, headline, lead\_paragraph or snippet section. Additionally, I removed articles pertaining to the words “Citi bike” and “Citi field” which are w relating to bike sharing and a news regarding a baseball park in New York City.

After identifying related articles, I went on to filter the dataset by sections. I manually scanned through the dataset and identified that sections "fashion & style", "arts", "books", "food", "sports", "travel", "the upshot", "blogs", "opinion", "u.s.", "your money" were unrelated to the business side of Citigroup.

Upon completion of preprocessing the Citigroup articles, the number of articles went from **24,516** to **108** articles.

Similarly for S&P related articles, I filtered out articles without terms "S.&.P", "S&P", "S.& P.", "S.& P.", "Standard & Poor" in the abstract, headline, lead\_paragraph or snippet section. After which I removed articles from the sections "fashion & style", "arts", "books", "food", "sports", "travel", "the upshot", "blogs", "opinion", "u.s.", "your money".

After preprocessing of S&P 500 articles, the number of articles went from **3,865** to **193** articles.

## Issues Encountered

# **Progress on Natural Language Processing**

## BERT

In order to obtain state of the art sentiment classification for both articles and tweets, I explored the use of Google’s Bidirectional Encoder Representation from Transformers (BERT) for my sentiment analysis. BERT is a new method of pre-training language representation and has been trained on a large text corpus.

I made use of Google’s pretrained BERT uncased model for my sentiment analysis. In order to finetune the BERT uncased model, I used a Kaggle dataset, Sentiment140. This dataset consists of 1.6million labelled tweets. The tweets are either labelled 0 for Negative or 4 for Positive.

When I first started finetuning the BERT model, I divided the dataset into 90:10 for training and testing. This means having 1,440,000 training data and 160,000 data in the testing set. This led to me encountering out of memory issues running the fine tuning on my local GPU. I then transferred the training onto Google Collaboratory, however, the finetuning process was too slow.

To overcome this issue, I decided to incrementally fine tune the BERT uncased model. Firstly, I shuffled and split the train and test set into 1,580,000 training data and 20,000 testing data. I did not follow the 80:20 rule in splitting the data in order prevent the out of memory issue when evaluating my model accuracy. Using a batch size of 32 and 3 epochs, I incrementally fine-tuned the BERT model by training on a smaller dataset, saving my trained model and incrementing the dataset and training the model again. Starting from 16,000 training data, each increment increases the training data by another 16,000 data records.

This is an evaluation of BERT sentiment classification on the 20,000 data test set before and after finetuning.

|  |  |
| --- | --- |
| **Evaluation of BERT Uncased model before finetuning** | |
| Evaluation Accuracy | 0.49345 |
| F1 Score | 0.6666666 |
| False Negative | 4734 |
| False Positive | 5397 |
| Loss | 0.6979323 |
| Precision | 0.49385726 |
| Recall | 0.5266 |
| True Negative | 4603 |
| True Positive | 5266 |

|  |  |
| --- | --- |
| **Evaluation of BERT Uncased model after finetuning on 80k dataset** | |
| Evaluation Accuracy | 0.81965 |
| F1 Score | 0.8165217 |
| False Negative | 1974 |
| False Positive | 1633 |
| Loss | 0.46343082 |
| Precision | 0.8309349 |
| Recall | 0.8026 |
| True Negative | 8367 |
| True Positive | 8026 |

## Sentiment Analysis on Twitter Tweets

Using the finetuned BERT Model, I classified each tweet into positive (1) and negative (-1). After which I summed all the tweets within the same day to get the overall sentiment for the day. For public holidays and weekends where the stock markets are not open, the day’s sentiment is carried forward to the following day.

This are the results of the sentiment analysis using BERT finetuned uncased model on Twitter Tweets.

## Sentiment Analysis on News Articles

To obtain the overall sentiment of a new article, sentiment classification is done on the abstract, headline, snippet and lead paragraph of the article, classifying each section as positive (1) or negative (-1). The overall sentiment of the article is obtained by summing the score for the abstract, headline, snippet and lead paragraph. If the summed value is greater than 0, it is classified as Positive, less than 0 would be classified as Negative, if it is equals to 0, the article is classified as Neutral sentiment.

These pie charts show the sentiment classification of articles for the different stocks.

## Issues Encountered

After performing sentiment classification on the news article dataset, I realized that there are significantly more articles classified with a negative sentiment. After manually checking through the dataset, I identified an issue with the sentiment classification of certain articles, primarily those that include news of other companies within the article.

The table below shows an example of an incorrectly classified news article:

|  |  |  |
| --- | --- | --- |
| Abstract | Summary: S&P 500 sets record high, *erases loss for 2014; J.C. Penney cutting staff, shutting some stores in new effort to rebound; Bank of America earnings inspire confidence; IMF's Lagarde warns against hasty Fed. Conway G. Gittens reports.* | -1 |
| Headline | **Record for S&P 500;** *J.C. Penney shrinking* | -1 |
| Snippet | **Summary: S&P 500 sets record high**, *erases loss for 2014; J.C. Penney cutting staff, shutting some stores in new effort to rebound; Bank of America earnings inspire confidence; IMF's Lagarde warns against hasty Fed. Conway G. Gittens reports.* | -1 |
| Lead Paragraph | **Summary: S&P 500 sets record high**, *erases loss for 2014; J.C. Penney cutting staff, shutting some stores in new effort to rebound; Bank of America earnings inspire confidence; IMF's Lagarde warns against hasty Fed. Conway G. Gittens reports.* | -1 |
| Sentiment | Negative | -4 |

The news article above is incorrectly classified as Negative even though the news on S&P 500 is that it is hitting record highs. This is because of the accompanying text with negative sentiments – “*J.C. Penney cutting staff, shutting some stores in new effort to rebound*”, “*J.C. Penney shrinking*”.

To overcome this incorrect sentiment classification, I intend to manually check and remove texts not relating to S&P 500 / Citigroup within the news article data set as the dataset is not large (193 articles for S&P 500, 108 articles for Citigroup).

# **Progress on LSTM Models**

To create my Long Short-Term Memory (LSTM) neural network models, I made use of TensorFlow Keras API along with Google Collaboratory to implement the models. All models consist of 1 input layer, 4 hidden layer and one output layer.

To begin I did feature engineering on the stock historical prices by comparing the accuracy of two models build using different number of features from the stock historical price data. The features available in the historical stock price dataset includes Date, Open, High, Low, Close, Adj Close and Volume.

The 1st Model uses the Closing price as the only feature when building the LSTM model

The 2nd Model uses the Closing, High and Low prices as features when building the LSTM model.

To create my stock price prediction model, I used Long Short-Term Memory (LSTM) neural network

# **Identifying Improvements**

## Further Data Cleaning for News Articles

## Further Fine Tuning

The current BERT model has only been finetuned with 80,000 labelled data. Further finetuning an evaluation of the model will be done.

## Further Tuning of Hyper Parameters