Predicting the Stock Market with Natural Language Processing

Author	V-Number
Tim Chan	803066
Adam Dubicki	799637
Shengzong He	785261
Lucas Sherman	711371
Haodong Tau	836169

Abstract

Predicting the stock market is a long existing analytical problem. This report details attempts to predict the Dow Jones Industrial Average (DJIA) using news headlines and natural language processing. The DJIA is a measure of major companies in the USA's industrial sector. The data set consists of news headlines from Reddit and 8 years of DJIA movement. Applying baseline supervised learning and natural language processing techniques, yielded a prediction accuracy of 57.9%. Further preprocessing techniques were then applied to yield a final prediction accuracy of 58.7% using logistic regression. Other data sets were experimented with, but performed poorly. This is a reasonable accuracy due to the quality of the data set and difficulty of the problem.

1 Introduction

The goal of this project is to use news headline data to predict the rise and fall of the United States industrial stock market.

To achieve this, labeled news headlines and supervised learning techniques will be used to create an accurate stock market classifier. This project is interesting for several reasons. Most of the data set pertains to international headlines. If a classifier can be created from the data, then it will illustrate how world news affects the US industrial sector. It will also reveal if popular aggregate news suffices to identify trends in the stock market.

Stock market prediction is a difficult challenge due to the many outside factors which affect it. Natural disasters, scandals, and product hype can all contribute to people buying or selling stock. Many of these factors can be captured by news aggregation. If the classifier built is sufficiently accurate, then it could be extended to make predictions on additional news aggregation sources.

1.1 Data Description

The data set for our chosen problem comes from Kaggle: a website which hosts various data mining and machine learning competitions. It was posted in August 2016 by user Aaron7sun ("Daily News for Stock Market Prediction", 2016). Reddit.com is a news aggregation website where users can vote on news headlines. The news headlines in the data set are from the 'world news' section. Reddit.com has a large amount of US visitors, resulting in many US specific headlines reaching the top 25 posts daily, despite the focus of the World News section being international headlines.

The data is packaged as a 6MB CSV file, each row containing news headlines and binary class. There are three files, RedditNews.csv, DJIA_table.csv and Combined_News_DJIA.csv. RedditNews.csv contains top 25 news headlines for a given day decided by the number of votes by Reddit users. DJIA_table.csv contains the stock data from Yahoo finance, and is based on the Dow Jones Industrial Average. ("DJI Historical Prices - Dow Jones Industrial Average Stock", 2017).

The DJIA is an indicator of the current state of the American industrial sector. It indicates how 30 large publicly owned companies based in the US (such as Microsoft, Verizon, Visa, IBM, etc.) have traded during a standard trading session in the stock market.

The Combined_News_DJIA.csv file gives the

top news headlines for every given day along with label entry saying whether the DJIA rose/stayed or fell. A row contains the Date, DJIA label, and then the top 25 headlines ordered with the number of votes. The combined file has 1990 rows, each corresponding to a day. Most of our processing was performed on Combined_News_DJIA.csv file.

In terms of data distribution the data set is slightly biased towards the DJIA rising (53.5% of the data is of class 1). Further on more data sets were added to try and boost the accuracy. These are detailed later in section 2.2.

1.2 Related Work

There are many ways that Natural Language Processing could assist in the movement of the stock market. For example, using the AlchemyData News API provided by IBM could allow machines to gather real-time intelligence on specific stocks, understand and respond to news events, monitor company sentiment and etc. (Sedlak, 2016).

There are also researchers that used Natural Language Processing and conducted a Textual Analysis for stock price prediction. The study lead by Stanford University introduced a system that forecasts company's stock price changes and classifies them into UP, DOWN and STAY. By building a well defined corpus that aligns the descriptions of financial events, the system's Unigram model could predict the stock trend with an accuracy of 54.4%. The low accuracy is because that newspaper articles reflect not only new information about the company, but also the perspective and options of third parties, which may require more time for the market to digest (Lee et al, 2014).

Two researchers one from Iona College and the other from University of Arizona worked together and performed textual analysis of the predicting the stock market using recent financial news. The researchers approached the problem using prediction with machine learning using financial news articles analysis where they researched with different textual representations, such as bag of words, noun phrases, and named entities. (Schumaker, R. P., & Chen, H. 2009).

The researchers investigated 9,211 financial news articles and over a five week period of 10,259,042 stock quotes that covered 500 of S&P stocks. They started their analyses of a stock price twenty minutes after a news article was released. A support vector machine (SVM) was used for discrete numeric prediction and models containing different stock-specific variables. They showed that the model that had both article terms and stock price at the time of the of the articles release was the closest prediction to the actual stock price. They achieved a prediction of the price moving in the same direction as the future price with a 57.1% accuracy and their highest return was 2.06% which used a simulated trading engine. They also investigated different textual representations and found that Proper noun scheme performs better the bag of words (Schumaker, R. P., & Chen, H. 2009).

Another report used sentiment analysis on Twitter posts to try and predict the daily closing values of the Dow Jones. The focus was on finding out whether the mood of the society was correlated to the stock price of the Dow Jones. Two different sentiment analysis tools were used: OpinionFinder and Google Profile of Mood States. OpinionFinder analyzed only a positive and negative moods, while Google Profile of Mood States would measure mood in term of 6 different dimensions. Correlation between mood and stock price was determined using Granger causality analysis. A Self-Organizing Fuzzy Neural Network was used to do the actual classification, which achieved an accuracy of 86.7%. Different moods were found to give different classification accuracies (Bollen et. al., 2010).

Researchers from the Malaysia has performed a study on text mining of news-headlines targeting specifically to the Foreign exchange (FOREX) market prediction. The study resulted the development of a multi-layer dimension reduction algorithm that consisted of three main layers: Semantic Abstraction Layer, Sentiment Integration Layer and Synchronous Targeted Feature Reduction. The proposed algorithm resulted with a model that has accuracy of up to 83.33% (Nassirtoussi et. al. 2015).

2 Methods

Below details the methods applied to try and yield the highest accuracy for predicting the stock market. Initially, a baseline was implemented based off a tutorial's code. Afterwards, the baseline was extended based on feedback from other peers.

2.1 Baseline

As part of the Kaggle competition, various tutorials are posted to help get users started with processing the data set, The highest voted tutorial was provided by Kaggle user Andrew Gélé. In the tutorial scikit-learns CountVectorizer is applied to transform the headlines into word vectors. The vectors contain the frequency of bi-grams in the headlines. The vectors are then processed and fitted with logistic regression. This approach yielded an accuracy of 56.3% (Gélé, 2016).

For Gélés approach, the data set was split by date such that all headlines from 2015 onward were test data. All other headline instances are part of the training set. This split was also used for our experiments for two reasons.

First, this aims to minimize multi-day headlines from being part of the training and test set. For example a week long event such as the Olympics could be highly correlated with the DJIA rising. If the data set is shuffled than there is a high probability of these headlines could be split across the training and test set. The classifier would then be training and testing on similar the same headlines, generating an optimistic accuracy. By splitting by date, related events shall be mostly segregated into the training and test sets. The only exception to this shall be events at the date boundary, which will unavoidably be split among the two sets.

Secondly, it is a very common split among other Kaggle competitors. By using the same split, our results are comparable to the results of other competitors.

To begin the experiments, a testing framework was built off the tutorial code in combination with other scikit learn libraries. The experiment setup in python aimed to create a highly iterable framework. The python file created was a preprocessor which would vectorize the data. The preprocessor also performed any data preparation steps such as stemming, stop-word removal and day offsetting.

Day offsetting is a minor improvement we experimented with, where the data is shifted forward or backward a certain amount of days. The unshifted data has headlines paired with stock results of that current day. This assumes that the headlines have made an impact on the current day. By offsetting, headlines for day i are paired with the binary class for day $i \pm o$, where o is the offset. The purpose behind this feature is two-fold. Examining positive shifts assumes that headlines have a latency towards affecting the a market. Negative shifts show how insider information and news leaks might allow for predictive trading.

Next a generic processor class was created. The processor takes as input a data set from the preprocessor, and a classifier from the python library scikit-learn. The processor splits the data into training and test sets, and trains with the training set. Finally, it forms predictions on the test set and outputs various metrics such as f-measure, accuracy, precision and recall.

After experimenting with several classifiers, the three best performing classifiers were logistic regression, SVM (linear kernal), and stochastic gradient descent. Through experimentation it was also determined that the a mix of bi-grams and tri-grams often yielded the highest results.

2.2 Extending the Baseline

After receiving feedback, the baseline method was extended in an aim to improve the accuracy.

- The world news data set does not always contain information relevant to the stock market. In an attempt to get more pertinent data, other subreddits (Economics, news (USA), and technology) were scraped using an API bot. Another experiment data set of headlines from the New York Times API was also scraped.
- The offset is an interesting feature which got minimal experimentation in the baseline. Only positive offsets were experimented with in the baseline. Additional offsets were attempted to see if future news insight had an affect on the market.

- While the headline date was used to split the data, it was never introduced as a feature. It was determined that the date could be pertinent if there were fluctuations around certain seasons or holidays. To accomplish this, the month and day integers were added as features to the vectors.

- Sometimes world events will span across multiple days of headlines. As discussed earlier, shuffling the data can cause cross contamination of headlines between the train and test data. Yet, it still may be important to reflect that events are trending. To represent this as a feature, a relation attribute was added to capture the similarity of a headline i to its surrounding headlines. It was calculated by taking the sum of cosine similarities between the headlines of day i with days $i+1, i+2, i-1$, and $i-2$. From observation this usually value ranged between 0 and 2. Headlines on the boundary (such as day 1) used a cosine similarity of 0 when calculating days outside of the valid range. When vectorizing the data for cosine similarity, a TFIDF vectorizer was used to give additional weighting to infrequent words.

A problem with Reddit data is that several of the subreddits were newer, or had less than 25 posts per day. This resulted in the new data sets having slightly different distributions and sizes. This limited the number of subreddits that could be scraped for data. For example, an attempt was made to scrape /r/stocks. Even at a small 10 headlines per day, only 667 days could be scraped due to the infrequency of posts.

Table 1: Data Set Info

Source	Rows	Headlines	% Class = 1
/r/Worldnews	1989	25	53.5%
/r/Economics	1981	10	53.3%
/r/Technology	1985	15	53.4%
/r/news(USA)	1984	15	53.6%
NY Times	1981	25	53.6%

Where /r/ means the data was scraped from Reddit's API. For the NY Times data set, the article headlines did not have a set rank for popularity. To generate a simple heuristic order, the headlines are sorted by length descending.

When forming the testing and training sets, the data was split by date as listed in section 2.1. This resulted in the test and training data sets having slightly different distributions.

Table 2: Class Distribution: Test and Training Sets

Source	% Class = 1 (Train)	% Class = 1 (Test)
/r/WorldNews	54.2%	50.8%
/r/Economics	54.0%	50.5%
/r/Technology	54.1%	50.8%
/r/news(USA)	54.2%	50.7%
NY Times	54.2%	50.8%

3 Results

Many of the tables share common feature names. For brevity some feature names have been abbreviated as follows.

LR	Logistic Regression
SVM	Support Vector Machine (Linear Kernel)
SGD	Stochastic Gradient Descent

3.1 Baseline Results

Table 3: Accuracy on World News (0 Day Offset)

	LR	SVM	SGD
Baseline	57.4%	57.1%	56.6%
+Stemming	56.6%	56.6%	55.8%
+Sentiment	57.4%	57.1%	54.8%
+Sentiment +Stemming	56.6%	56.6%	57.9%

Table 4: Metrics for SGD on World News with Stemming and Sentiment

Class	Precision	Recall	F1-Score
0	0.61	0.40	0.48
1	0.56	0.76	0.65

Table 5: Most Positive Features for SGD on World News with Stemming and Sentiment

Rank	N-Gram
1.	(west, bank)
2.	(it, has)
3.	(to, the)
4.	(right, to)
5.	(what, the)
6.	(in, China)
7.	(teenagers, in, many)
8.	(the, first)
9.	(and, other)
10.	(in, south)

3.2 Baseline Discussion

The baseline results showed some promise that a reasonable classifier could be built from the World News data set. Simply changing the the use of bi-grams to a combination of bi-grams and tri-grams boosted the tutorials code from 56.3% to 57.4% with logistic regression (Table 2).Additional features such as stemming and and sentiment did not boost the accuracy on their own. Yet a combination of both helped SGD yield the most accurate World News classifier at 57.9%.

One of the disappointing results is that SVMs did not perform better than the other classifiers. SVMs are often good at classifying on high dimensional and sparse data sets. Unfortunately it performed only as well or worse than logistic regression and SGD. More complex classifiers such as neural networks were experimented with but had their short comings. Large neural networks require a lot of computational power. This meant the neural networks we could create were limited in size due to the potential for memory errors. Secondly, our data set is small (less than 2000 rows of data). Neural networks generally require larger data sets to be trained properly. Using duplicate data to artificially inflate the size of the data set could be performed. Though this was not attempted, and neural networks were not experimented with any further for this project.

The metrics from our best classifier were extracted to form Table 4. The precision for predicting class 0 is better than class 1, but has much lower recall. The reason for this is most likely due to the bias in the data set. The classifier predicted 1 by default which captured a lot of the True positives, but struggled to differentiate them from false positives. While class 0 was harder to detect (recall of 0.40) it ended more precise (precision of 0.61). This indicates that there are some features which strongly indicate that a class was 0, but are infrequent within the data set.

Table 5 displays the N-Grams which had the highest weighting in the classifier. They indicate which N-Grams are most likely to make the classifier predict 1. The most interesting of these is 'West Bank' which appeared 219 times in the data set. West bank is a region in western Palestine. Other interesting such as 'in China' and 'in South' indicate that headlines from around the world can impact the Dow Jones. The other N-grams are less interesting because they contain stop words. Removing stop words decreased the accuracy during experimentation.

3.3 Extended Results

Table 6: Accuracy on World News (0 Day Offset) Extended

	LR	SVM	SGD
+Dates	57.7%	56.3%	49.4%
+Relation	57.7%	56.6%	55.0%
+Sentiment +Stemming +Dates +Relation	56.1%	56.1%	51.8%

Table 7: Accuracy on Economics (0 Day Offset)

	LR	SVM	SGD
Baseline	50.5%	50.2%	47.3%
+Stemming	51.3%	51.0%	48.9%
+Sentiment	50.5%	50.3%	47.3%
+Sentiment +Stemming	51.3%	51.0%	49.4%
+Dates	49.2%	50.8%	49.7%
+Relation	50.8%	50.3%	50.2%
+Sentiment +Stemming +Dates +Relation	49.2%	51.9%	50.8%

Table 8: Accuracy on Technology (0 Day Offset)

	LR	SVM	SGD
Baseline	52.4%	51.3%	49.4%
+Stemming	51.6%	50.8%	48.1%
+Sentiment	52.4%	51.3%	52.1%
+Sentiment +Stemming	51.6%	50.8%	49.2%
+Dates	52.4%	51.6%	48.9%
+Relation	52.4%	51.3%	49.5%
+Sentiment +Stemming +Dates +Relation	50.7%	51.3%	47.9%

Table 9: Accuracy on News (USA) (0 Day Offset)

	LR	SVM	SGD
Baseline	51.5%	49.9%	48.0%
+Stemming	49.9%	49.0%	51.2%
+Sentiment	51.2%	49.9%	50.4%
+Sentiment +Stemming	49.6%	48.8%	49.3%
+Dates	50.1%	50.7%	51.4%
+Relation	51.5%	49.9%	49.3%
+Sentiment +Stemming +Dates +Relation	49.8%	49.6%	49.6%

Table 10: Accuracy on New York Times (0 Day Offset)

	LR	SVM	SGD
Baseline	50.5%	51.0%	50.0%
+Stemming	49.7%	51.0%	50.8%
+Sentiment	50.5%	51.1%	49.7%
+Sentiment +Stemming	51.8%	51.6%	49.2%
+Dates	50.1%	50.7%	51.5%
+Relation	50.2%	51.2%	51.6%
+Sentiment +Stemming +Dates +Relation	49.8%	49.6%	49.6%

Table 11: Metrics for Logistic Regression on World News with Date (-1 Day Offset)

Class	Precision	Recall	F1-Score
0	0.69	0.30	0.41
1	0.56	0.87	0.68

Table 12: Most Positive Features for Logistic Regression on World News with Date (-1 Day Offset)

Rank	N-Gram
1.	(right, to)
2.	(and, other)
3.	(the, first)
4.	(set, to)
5.	(in, China)
6.	(will, be)
7.	(to, the)
8.	(New, Zealand)
9.	(found, in)
10.	(after, the)

Table 13: Most Negative Features for Logistic Regression on World News with Date (-1 Day Offset)

Rank	N-Gram
1.	(in, Gaza)
2.	(to, kill)
3.	(up, in)
4.	(with, Iran)
5.	(Bin, Laden)
6.	(people, are)
7.	(there, is)
8.	(fire, on)
9.	(to, help)
10.	(10, 000)

3.4 Extended Discussion

After gathering new data sets, all of our preprocessing features were attempted to try and yield a higher accuracy than the baseline. From Table 6-10, none of the new features were able to yield a higher accuracy than 57.9% on any of the data sets. Disappointingly, all of the new data sets performed poorly with a 0 day offset. The new relation and data feature did not boost the accuracy by too much. From table 6 it did not hinder the prediction. Interestingly, the date feature seemed to throw off SGD, dropping the accuracy. This is most likely due to a preprocessing mistake, as those features were not normalized. This mistake however did not significantly lower the accuracy on LR nor SVM.

Upon seeing that the new data and features were unable to boost the accuracy, various shifted day offsets were performed (see Appendix A-E). None of the day offsets helped the alternative data sets exceed the results of the baseline (see Appendix B-E). Excitingly the offsets were able to make a reasonable impact on the World News data set.

All of the data shifts were able to yield results as good or better than the 0 day offset. The best of these shifts was the -1 day offset which yielded an accuracy of 58.7% using LR with the date feature (see Appendix A, Table 15). From that configuration the classifier metrics were extracted to form Table 11. Comparing to the baseline results (Section 3.1, Table 4), some interesting changes occurred. The recall for class 0 decreased significantly 10%, yet the precision increased by 8%. For class 1, the precision stayed the same at 56%, but the recall increased by 11%.

Further drilling down into the classifier yielded tables 12 and 13. Once again, many stop-word sentences appeared as significant features. More interesting though is that many locations appeared as significant N-Grams. China, New Zealand are associated with the DJIA rising/staying the same while Gaza and Iran are associated with the DJIA falling. Table 13 yielded many interesting features, as most of the N-grams are associated with violence and war (to kill, Bin Laden). These kind of results give reason that international headlines can give some insight the US stock market.

Initially it was hypothesized that our classifiers had approached an accuracy limit on the World News data set. This incentivized gathering other data sets in hopes of bypassing this limit. Generating new data sets from reddit and the New York Times' API yielded low accuracy.

It is interesting that a -1 day offset performed well for the /r/worldnews. A possibility for a -1 offset being beneficial would be if insider trading is very prevalent and causes changes in stock price before there is any news on it. Another possibility is that stock prices are affected more heavily by rumours, rather than news.

4 Conclusion

It is much more difficult than we anticipated to try and predict the stock market based off of news headlines. We were able to achieve a final accuracy of 58.7%. While not an amazing accuracy, it is fairly strong considering the difficulty of the problem, and weaknesses of the reddit data set.

Stemming and sentiment from the baseline gave a decent accuracy boost of around 1.5%. Disappointingly, the extended methods were only able to boost the accuracy at best from 57.8% to 58.7%. The date attribute seemed actually helped in most cases which indicates that the time of year be a key attribute. Our most complex feature - the relation attribute - did not provide much of an accuracy boost. This is most likely due to the size of the text that the cosine similarity was taken between. Since the features contained 25 headlines, the bi-gram overlap was usually quite small. Shifting the data set overall helped, with a -1 offset being a helpful preprocessing step.

One of the interesting results is that the world news data set ended up being the most accurate. This went against our initial ideas that other subreddits may contain more pertinent information. Other data sets contained a mix of headlines and personal user content. The personal content usually consisted of user discussions, or question-answer type posts. These types of posts added noise to the data set. While /r/worldnews did not always contain pertinent information, all of the posts are at least news headlines. The American news subreddit - /r/news - underperformed and was not able to beat the world new data set.

The most interesting result is the number of locations which appeared as important features. Places such as Iran, Gaza, New Zealand, and China appeared as important. This gives some validity to world news being able to affect the US stock market.

Throughout this project we learned many things:

- Data quality is sometimes non-intuitive. Data sets which we thought might be more accurate ended up being very inaccurate. While these data sets could be useful by applying other methods, they were poor in our application. Sometimes data sets which could have been useful (such as /r/stocks) were not popular enough to form a data set. If we were to restart this project, we would begin by looking for other ways to aggregate news. The restriction of reddit is that data sets often end up being small. Twitter might be a better data set due to the hash tag, allowing for large amounts of pertinent data to be scrapped.

- Preprocessing steps are sometimes detrimental. Originally stop-word removal was included, but it always lowered the accuracy. This was unintuitive because stop-words do not contain important information about the topic. Stop-words actually ended up being important features in the baseline and extended classifiers. Meanwhile, stemming

and sentiment usually provided a decent accuracy boost. Stemming makes sense because it shrinks the vector space for words. Sentiment is interesting because our intuition is that most headlines would be neutral. Yet a combination of these two yielded the highest accuracy in the baseline results 57.9 (Table 3, section 3.31). While the date attribute was helpful for SVMs and logistic regression, it often ruined the accuracy of SGD. We believe this is due to not normalizing the date attributes. While every other features was in the range of 0 and 1, the date attributes ranged from 1 to 31.

- Predicting the stock market is hard. There are many outside features not captured by the headlines which could affect the trends in the market. Even when subjects are trending, the data set had to be split in a way that similar headlines were kept separate. We attempted to capture the trends by using a relation feature and checking for similar subjects in surrounding headlines. This only gave a marginal accuracy boost. More complex classifiers such as neural networks still may be able to classify the stock market. Unfortunately they require more data and computational power than we had at our disposal.

4.1 Future Work

If the project is to be continued, it would be interesting to gather a larger data set.. With a larger data set, it should be viable to use some sort of LSTM. Since, LSTMs are good at remembering things for long or short periods of time. This makes them useful in time series prediction. It should be able to use this property to predict short and long term trends within the market. Also, for actually making money off the stock market, it would be necessary to use some sort of Named Entity Recognition system to recognize key stocks. It may also be worth experimenting with more dimensions in the sentiment analysis. Analyzing the quality of the news headlines could also help, by using stance detection and click-bait detection to see how reliable certain articles are before their headlines are used.

Currently our data set has only two classes: DJIA rise/stayed the same and DJIA falls. It could be worth investigating if adding a third class would help. In this case the class 1 (rise/stayed the same) would be split. There is a danger of lowering the accuracy, because this would mean the classifier would have to predict the classes more clearly.

4.2 Societal Implications

It's difficult to imagine that a perfect stock market classifier would ever be possible due to the many factors that have to be taken into account. Assuming that it could be perfected, it would have some significant societal implications. If a group is able to secretly perfect this classifier then they would be able to make large profits off of it. Considering that it would be likely that this group was wealthy enough to put a lot of it into research, then it would mean that the rich were getting richer. But if the group did share this information, then it would quickly be adapted by many people. It would mean that only the most promising stocks would be heavily invested in, and their stock price would increase sharply, guaranteeing that the classifier is correct. So a classifier which is good enough may become self-fulfilling. An important thing to note though, is that if the stock market classifier worked off of news, then there would likely be a significant industry based on generating fake news to mess up competitors. A side effect of this would be it would be harder for people to know which news sources are reliable.

4.3 Final Thoughts

Overall the project was a success. It was interesting and unexpected for locations outside the USA to be such important headline features. 58.7% accuracy is not outstanding, but it was able to exceed a couple of the related work papers. We are still hopeful that this project can someday be continued, and experimented with other more complex classifiers such as neural networks.

4.4 Contributions

The original plan was to divide the main body of work into 3 parts: preprocessing, processing and results gathering. Each team member was to work on two of 3 parts. As the project progressed, this structure fell apart. In the end, team members contributed to each of the three sections as well as milestone items.

- Tim: Helped with proposal, wrote the original pre-processor, helped with processing the data and trying different classifiers, worked on presentation slides and presented, helped with writing final report.

- Adam: Wrote the processor code. Added additional features to the preprocessor for the date and relation attributes. Worked on the presentation slides and presented. Helped writing the final report, proposal and presentation. Wrote the python tool to fetch data from Reddit. Scrape the Economics and News (USA) data sets.

- John: Helped with proposal, helped with pre-processor. Wrote demo code for presentation. Worked on generating dataset from New York Times. Assist in processor code. Help writing the final report.

- Tao: Helped with proposal. Worked on generating dataset from the technology subreddit. Helped writing the final report.

- Lucas: Helped with proposal.Worked on the presentation slides and presented.Helped writing the final report.

5 Appendix A. Shifted WorldNews Data

Table 14: Accuracy on World News (-2 Day Offset)

	LR	SVM	SGD
Baseline	54.8%	54.5%	54.0%
+Stemming	54.8%	55.6%	55.0%
+Sentiment	54.8%	54.5%	55.0%
+Sentiment +Stemming	54.8%	55.8%	55.6%
+Dates	57.4%	57.1%	51.1%
+Relation	56.6%	56.6%	55.6%
+Sentiment +Stemming +Dates +Relation	56.9%	55.6%	52.9%

Table 15: Accuracy on World News (-1 Day Offset)

	LR	SVM	SGD
Baseline	58.2%	57.7%	53.4%
+Stemming	56.3%	56.3%	57.4%
+Sentiment	58.2%	57.7%	57.1%
+Sentiment +Stemming	56.3%	56.3%	57.4%
+Dates	58.7%	57.4%	51.0%
+Relation	58.0%	57.7%	55.0%
+Sentiment +Stemming +Dates +Relation	57.4%	56.3%	50.0%

Table 16: Accuracy on World News (+1 Day Offset)

	LR	SVM	SGD
Baseline	57.9%	57.4%	54.7%
+Stemming	56.6%	56.3%	55.0%
+Sentiment	57.9%	57.6%	54.4%
+Sentiment +Stemming	56.6%	56.3%	55.3%
+Dates	58.5%	57.4%	51.1%
+Relation	58.2%	56.9%	55.3%
+Sentiment +Stemming +Dates +Relation	56.6%	55.8%	51.1%

Table 17: Accuracy on World News (+2 Day Offset)

	LR	SVM	SGD
Baseline	57.1%	55.2%	57.9%
+Stemming	57.1%	55.3%	57.9%
+Sentiment	56.6%	55.5%	54.5%
+Sentiment +Stemming	57.1%	55.3%	57.4%
+Dates	56.3%	55.8%	53.7%
+Relation	56.3%	55.6%	54.2%
+Sentiment +Stemming +Dates +Relation	57.4%	55.3%	52.3%

6 Appendix B. Shifted Economics Data

Table 18: Accuracy on Economics (-2 Day Offset)

	LR	SVM	SGD
Baseline	50.8%	50.8%	47.9%
+Stemming	52.1%	51.3%	48.1%
+Sentiment	50.8%	50.8%	47.9%
+Sentiment +Stemming	52.1%	51.4%	49.4%
+Dates	50.8%	50.3%	49.7%
+Relation	50.8%	50.8%	46.25%
+Sentiment +Stemming +Dates +Relation	51.3%	51.6%	49.5%

Table 19: Accuracy on Economics (-1 Day Offset)

	LR	SVM	SGD
Baseline	50.5%	49.7%	44.1%
+Stemming	49.4%	49.2%	44.1%
+Sentiment	50.5%	50.0%	49.4%
+Sentiment +Stemming	52.1%	52.4%	47.3%
+Dates	51.1%	52.1%	50.5%
+Relation	51.6%	51.6%	48.4%
+Sentiment +Stemming +Dates +Relation	52.4%	50.8%	49.4%

Table 20: Accuracy on Economics (+1 Day Offset)

	LR	SVM	SGD
Baseline	49.2%	51.8%	46.5%
+Stemming	51.6%	51.8%	44.9%
+Sentiment	52.4%	51.8%	46.5%
+Sentiment +Stemming	49.2%	49.2%	45.7%
+Dates	50.0%	50.5%	47.6%
+Relation	50.3%	50.5%	49.4%
+Sentiment +Stemming +Dates +Relation	49.5%	51.3%	49.4%

Table 21: Accuracy on Economics (+2 Day Offset)

	LR	SVM	SGD
Baseline	49.4%	49.2%	46.5%
+Stemming	49.7%	49.4%	45.2%
+Sentiment	49.4%	49.2%	47.7%
+Sentiment +Stemming	50.0%	50.0%	48.9%
+Dates	48.9%	50.3%	50.3%
+Relation	49.2%	49.2%	49.4%
+Sentiment +Stemming +Dates +Relation	49.2%	51.6%	51.3%

7 Appendix C. Shifted Technology Data

Table 22: Accuracy on Technology (-2 Day Offset)

	LR	SVM	SGD
Baseline	51.1%	50.5%	51.1%
+Stemming	51.3%	50.8%	50.3%
+Sentiment	51.1%	50.5%	51.3%
+Sentiment +Stemming	51.3%	50.8%	51.9%
+Dates	50.8%	51.3%	49.7%
+Relation	51.3%	50.5%	48.9%
+Sentiment +Stemming +Dates +Relation	51.1%	52.3%	49.3%

Table 23: Accuracy on Technology (-1 Day Offset)

	LR	SVM	SGD
Baseline	50.8%	50.5%	53.5%
+Stemming	52.1%	52.7%	50.8%
+Sentiment	50.8%	50.5%	52.4%
+Sentiment +Stemming	52.1%	52.7%	49.5%
+Dates	51.1%	53.0%	49.2%
+Relation	50.8%	50.3%	50.0%
+Sentiment +Stemming +Dates +Relation	53.2%	53.7%	48.9%

Table 24: Accuracy on Technology (+1 Day Offset)

	LR	SVM	SGD
Baseline	51.6%	51.3%	50.3%
+Stemming	52.7%	51.3%	50.8%
+Sentiment	51.6%	51.3%	48.1%
+Sentiment +Stemming	52.7%	51.3%	49.2%
+Dates	52.3%	52.7%	48.4%
+Relation	52.4%	51.3%	50.0%
+Sentiment +Stemming +Dates +Relation	51.9%	51.6%	48.1%

Table 25: Accuracy on Technology (+2 Day Offset)

	LR	SVM	SGD
Baseline	51.6%	51.9%	49.7%
+Stemming	52.4%	51.3%	48.9%
+Sentiment	51.6%	51.9%	49.2%
+Sentiment +Stemming	52.4%	51.6%	48.4%
+Dates	51.3%	51.1%	47.1%
+Relation	51.6%	51.9%	49.2%
+Sentiment +Stemming +Dates +Relation	51.3%	50.5%	49.2%

8 Appendix D. Shifted News (USA) Data

Table 26: Accuracy on News (USA) (-2 Day Offset)

	LR	SVM	SGD
Baseline	50.4%	49.6%	49.6%
+Stemming	51.2%	50.1%	51.4%
+Sentiment	50.4%	49.6%	49.1%
+Sentiment +Stemming	51.2%	50.1%	48.2%
+Dates	51.7%	49.9%	50.7%
+Relation	50.4%	49.9%	50.9%
+Sentiment +Stemming +Dates +Relation	51.5%	50.1%	49.3%

Table 27: Accuracy on News (USA) (-1 Day Offset)

	LR	SVM	SGD
Baseline	52.0%	49.9%	50.1%
+Stemming	49.6%	49.6%	50.4%
+Sentiment	52.0%	49.7%	53.1%
+Sentiment +Stemming	49.3%	49.6%	51.5%
+Dates	52.8%	52.3%	50.4%
+Relation	51.7%	49.9%	49.3%
+Sentiment +Stemming +Dates +Relation	50.4%	50.4%	48.8%

Table 28: Accuracy on News (USA) (+1 Day Offset)

	LR	SVM	SGD
Baseline	49.8%	48.5%	50.4%
+Stemming	49.0%	47.7%	47.7%
+Sentiment	49.6%	48.5%	49.3%
+Sentiment +Stemming	48.8%	47.7%	49.3%
+Dates	49.1%	50.4%	47.7%
+Relation	49.9%	49.3%	48.5%
+Sentiment +Stemming +Dates +Relation	49.3%	48.8%	46.9%

Table 29: Accuracy on News (USA) (+2 Day Offset)

	LR	SVM	SGD
Baseline	48.8%	51.5%	50.9%
+Stemming	49.9%	49.1%	51.2%
+Sentiment	48.8%	51.5%	50.4%
+Sentiment +Stemming	49.9%	48.8%	45.3%
+Dates	49.3%	51.2%	45.6%
+Relation	48.8%	51.4%	49.6%
+Sentiment +Stemming +Dates +Relation	49.3%	48.5%	45.9%

9 Appendix E. Shifted New York Times Data

Table 30: Accuracy on New York Times-2 Day Offset)

	LR	SVM	SGD
Baseline	53.2%	53.2%	49.2%
+Stemming	50.5%	50.4%	48.1%
+Sentiment	53.2%	53.2%	49.2%
+Sentiment +Stemming	50.5%	50.0%	50.3%
+Dates	53.2%	53.2%	51.1%
+Relation	53.4%	52.9%	50.3%
+Sentiment +Stemming +Dates +Relation	51.6%	53.2%	49.2%

Table 31: Accuracy on New York Times (-1 Day Offset)

	LR	SVM	SGD
Baseline	50.8%	50.3%	50.3%
+Stemming	50.3%	50.3%	50.3%
+Sentiment	50.8%	50.3%	51.1%
+Sentiment +Stemming	50.3%	50.3%	47.9%
+Dates	52.8%	52.3%	50.4%
+Relation	50.8%	50.3%	49.2%
+Sentiment +Stemming +Dates +Relation	50.4%	50.4%	48.8%

Table 32: Accuracy on New York Times (+1 Day Offset)

	LR	SVM	SGD
Baseline	53.7%	52.9%	51.1%
+Stemming	50.3%	50.8%	50.0%
+Sentiment	53.7%	52.9%	50.5%
+Sentiment +Stemming	50.3%	51.1%	50.0%
+Dates	49.1%	50.4%	47.7%
+Relation	52.9%	52.9%	50.8%
+Sentiment +Stemming +Dates +Relation	49.3%	48.8%	46.9%

Table 33: Accuracy on New York Times (+2 Day Offset)

	LR	SVM	SGD
Baseline	52.1%	49.7%	50.3%
+Stemming	49.7%	50.5%	49.2%
+Sentiment	52.1%	49.7%	51.9%
+Sentiment +Stemming	49.7%	50.5%	50.0%
+Dates	49.3%	51.2%	45.6%
+Relation	52.1%	49.5%	49.7%
+Sentiment +Stemming +Dates +Relation	49.3%	48.5%	45.9%

10 References

Daily News for Stock Market Prediction. Kaggle.com. Retrieved 14 May 2017, from https://www.kaggle.com/aaron7sun/stocknews

Bollen, Johan & Mao, Huina & Zeng, Xiao-Jun. (2010). *Twitter Mood Predicts the Stock Market.* Journal of Computational Science. 2. . 10.1016/j.jocs.2010.12.007. Retrieved 22 July 2017, from https://arxiv.org/pdf/1010.3003.pdf

DJI Historical Prices — Dow Jones Industrial Average Stock - Yahoo Finance. (2017). Finance.yahoo.com. Retrieved 14 May 2017, from https://finance.yahoo.com/quote/%5EDJI/history?p=%5EDJI

Gélé, Andrew. (2016, November). *OMG! NLP with the DJIA and Reddit! — Kaggle.* Retrieved 14 May 2017, from https://www.kaggle.com/ndrewgele/omg-nlp-with-the-djia-and-reddit

Lee, H., Surdeanu, M., MacCartney, B., & Jurafsky, D. (2014, May). *On the Importance of Text Analysis for Stock Price Prediction.* In LREC (pp. 1170-1175). Retrieved 14 May 2017, from http://www.lrec-conf.org/proceedings/lrec2014/pdf/1065_Paper.pdf

Nassirtoussi A.K., Aghabozorgi S., Wah T.Y., Ngo D.L.C. (2015). *Text mining of news-headlines*

for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. Expert Systems with Applications, Volume 42, Issue 1, 2015, Pages 306-324, ISSN 0957-4174. Retrieved 22 July 2017, from http://www.sciencedirect.com/science/article/pii/S0957417414004801

Schumaker, R. P., & Chen, H. (2009). *Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 12.* Retrieved July 15 2017, from http://dl.acm.org/citation.cfm?id=1462204

Sedlak, M. (2016). *How Natural Language Processing is transforming the financial industry - IBM Watson.* IBM. Retrieved 14 May 2017, from https://www.ibm.com/blogs/watson/2016/06/natural-language-processing-\transforming-financial-industry-2/

Trastour, S., Genin, M., & Morlot, A. (2016).*Prediction of the crude oil price thanks to natural language processing applied to newspapers.* Retrieved 14 May 2017, from http://cs229.stanford.edu/proj2016/report/GeninMorlot\Trastour-PredictionOfTheCrudeOilPrice\ThanksToNLPAppliedToNewspapers-report.pdf