Stock Market Prediction using Daily News Headlines

Article *in* SSRN Electronic Journal · January 2020

DOI: 10.2139/ssrn.3685530

3 authors, including:

Ali Hassanzadeh Kalshani
The University of Manchester
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Stock Market Prediction using Daily News Headlines

Ali Hassanzadeh
Merage School of Business
ali.h@uci.edu

Ahmad Razavi
Donald Bren School of CS
arazavi@uci.edu

Reza Asadi
Donald Bren School of CS
rasadi@uci.edu

Abstract

In this work, we develop a model for stock market forecasting using *News Headlines*. The main step of the work is to pre-process daily news, as well as Opinion Finding method to generate features for the prediction problem. The target variable is a binary variable which takes value of 1, when the *Dow Jones Industrial Average (DJIA)* increases, and zero otherwise. A deep learning model with *Long Short-Term Memory (LSTM)* architecture to improve the prediction of the changes in DJIA.

As a baseline, a time series data of DJIA is used without daily news. Then, we develop a model of time series prediction along with *N-Gram* of daily news. We discuss how N-gram model can help to improve the baseline. In the main pre-processing task, we develop a sentiment analysis (using polarity lexicons) with the methods proposed by (Riloff and Wiebe, 2003) and (Pappas et al., 2013). This model assigns weights as *sentiment polarity* to all the news headlines. We also find the scores for degree of *subjectivity* and *objectivity* in news headlines, and we use them as additional features in our model.

In experiments, an LSTM model is developed. A grid search is used to find the best architecture along with the effect of generated features from text pre-processing and Opinion Finding on time series forecasting.

1 Introduction

Predicting multi-variate time series, in general, has been one of the difficult tasks in machine learning, and surely predicting stock market prices as an especial case and a nice example of a multi-variate time series problem, is one of the interesting and fruitful research areas. Of course a lot of factors contribute to the final stock prices that we see on a daily basis, and one cannot nail it down to only few variables which explain almost all the variability in prices. One thing that everybody in stock market community agrees is the fact that politics and what is happening in the world do have a significant impact on stock prices. Now, the task at hand is, given news headlines on a daily basis, can we come up with a methodology to predict an Dow Jones Index.

A lot of studies have been done, especially during recent years, addressing the question of whether we can use text (e.g., news, social media, search engine results, etc.) in order to predict what is going to happen to stock market (Bollen et al., 2011), (Ding et al., 2015), (Ding et al., 2015). People have discovered connection between huge changes in prices and changes in the frequency of financial keywords being searched on search engines. Some other studies focused on the impact of social media, especially Twitter, on the major changes in stock market explaining, indexes. Even though all of these studies are very nice examples of analyzing the stock market prices, combining several ideas seems to be missing from the number of papers that we studied.

1.1 Data Description

We use a dataset containing news headlines and stock market index avaiable in Kaggle[1]. It contains the top 25 daily news headlines every single weekday starting from August 8, 2008, all the way till June 2016. Daily values for Dow-Jones Industrial Average (DJIA) is also available. The news headline is selected from Reddit as a world daily news, containing economic and non economic news.

1.2 Methodology

The main goal of the project is to apply opinion finding and other text preprocessing methods on the dataset to generate new features for stock

[1] Kaggle: Daily News for Stock Market Prediction

market forecasting. This analysis represent how these methods are helpful in stock market forecasting. As a baseline model, we use time series data of Dow Jones Index. Then, we develop *n-gram* model on daily news. Also, we use additional features obtained from *sentiment analysis*, which fully described and analyzed. In next analysis, we examined objectivity and subjectivity. We come up with two features (average of subjectivity or objectivity of news in each day), between 0 and 100. These two numbers, obviously tend to complement each other, meaning that at any day, the overall news tend to have high subjectivity value (e.g., 78.50), the score for objectivity would be lower (e.g., say 21.50). In last pre-processing section, we do *opinion finding (i.e., mood mining)*, in which we perform a sequence labeling of the new headline text, and we assign a number known as *sentiment polarity* to each token. Then, we take the average value for each day, and by normalizing it, we have a value between 0 and 100 for each day, as to how much *positive*, *negative*, or *neutral* that day was regarding the daily news. A similar work has been done on a Twitter corpus which apparently is the most cited paper in this stream of research.

Finally, we develop an *LSTM-based* deep learning neural network, and generating good set of features, we would be able to strengthen the model with a descent margin. We evaluate the modified *LSTM* model (with added set of features from sentiment analysis and mood mining) on test data, to see how good our model can predict DJIA, and if not, what could be a plausible theory as to why they do not work.

2 Related Work

There are quite a lot of recent studies on the stock market prediction using text analysis. For instance, (Ding et al., 2015) have focused on extracting certain *events* from the news text, and building predictions based on these events. They use a two-stage framework in which they first extract events from the text, and then they use *Convolutional Neural Network (CNN)* to predict the stock prices. One shortcoming of this paper might be the fact that they do not use RNN which is much well-suited to the problems where previous (going back in time slots) response variables do play an important role in current response variable.

As another related work, (Zhai et al., 2007) combine news headlines with technical indicators in order to do the prediction. The main idea in this paper is that one source of information as an input to stock price prediction is not enough, and a lot of papers neglect the influence of mass media on the behavior of the investors. In their proposed methodology, they use multiple *Support Vector Machine (SVM)* models to predict the stock prices.

Another recent work has been done by (Li et al., 2014), and they focus specifically on the *sentiment analysis*, and on working on the text analysis in order to predict future market situation. They use *Harvard psychological dictionary* and *Loughran-McDonald financial sentiment dictionary* in order to construct a sentiment space to work with.

In the paper by (Nguyen et al., 2015)), the argument is that most of text analysis works around the topic of stock market prediction, consider overall mood and sentiment of the text, not the most specific ones related to each company. In this work, authors consider topics related to the company and they use the keywords that are related to the company itself to determine whether that news or a post on social media has positive or negative impact on the stock price of this specific company.

The work by (Bollen et al., 2011) is a nice and relatively well-known paper where the authors argue that general mood of a social media does have an effect on economic factors, like stock market indicators (e.g., DJIA). They use two mood tracking tools, namely *OpinionFinder* (which only determines whether a tweet is positive or negative), and *Google-Profile of Mood States (GPOMS)* (which assigns a mood to a tweet out of 6 possible states). They use the mood that the time series generates, as an input for their prediction model.

As an explanatory study about the nature of variability in stock market prices, we can refer to the work by (Moat et al., 2013), in which they have found evidence of close connection between sudden changes in English Wikipedia article views of *financial topics* and sudden drops or raises in stock market indicators (e.g., DJIA).

As explained nicely in the work by (Pappas et al., 2013), we have mainly two categories of the methods in *sentiment polarity*, namely *lexicon-based* and *classification-based*. The authors use a lexicon-based algorithm to calculate the *subjectivity* and *objectivity* of the input text. They look

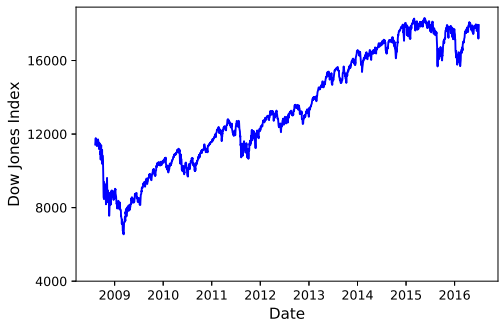


Figure 1: The value of Dow Jones Index over eight years

for relevant text in the web, targeting specific genres (e.g., news, blogs, discussion forums) that are highly probable to contain *opinionated text*. In the huge database of text on web, they search user-generated content around the topics that they have found, and they somehow use a network of relevant text to come up with a score (for both subjectivity and objectivity) for the original text input. We are using their methodology to obtain an average score of subjectivity and objectivity for each day.

3 Approach

We consider a Time Series of DJIA, without news headlines, as our *Baseline Model*. We show that using the stock market values, the performance of the prediction is not promising. To improve this performance, we pre-process the input text, and use it as an input for our *LSTM* model. In the following, we explain baseline model, the two main pre-processing text analysis, and then the main LSTM model.

3.1 Time Series Analysis (baseline)

The original dataset is a time series dataset of DowJones Index. It consists of Open, Close, Low, High, Adj-close, and the Volume. As it is shown in Figure (1), the DJI is a continuous volume. In Kaggle data set, to simplify this problem to classification, the output for everyday is considered as 0/1 for the days that have decrease and decrease/increase compared to the previous day.

To apply time series analysis and improve the baseline model, we used the continuous volume data of data set to predict y_{t+1}, given the input features of $\{x_1, \ldots, x_t\}$, where $x_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$. A time window is used to consider part of past x values as input of the model.

3.2 Text Feature Engineering

In order to generate more features from raw text as numerical values, we have considered various parameters including:

- Linguistic features: number of words, characters, unique words, stop words, and also average length of each headline
- Decomposed *TF-idf* vectors using SVD
- Decomposed embedding vectors
- N-gram model

These are basic features generated from daily news. These features are examined in section (4.2) and (4.3).

3.3 Sentiment Analysis

The *subjectivity*, *objectivity*, and *polarity* are important parts of the meaning of each sentence. Subjectivity and objectivity means how much a sentence is influenced with feelings, emotions, personal opinions, etc. (in the case of subjectivity), and how much it is drawn from the actual facts (in the case of objectivity). It is possible to *measure* these features and use them in an NLP model. We use the algorithms proposed by (Pappas et al., 2013) and (Riloff and Wiebe, 2003) to generate these new sets of features for all of the news headlines of each day. Apart from these two interesting features, we calculate average weight associated with *positive*, *negative*, and *neutral* polarity measures for all the news headlines of each weekday.

For instance, for the very first day in our dataset (i.e., August 8, 2008), overall news gave us $(75, 25)$ as subjectivity and objectivity measures, respectively. For the same day, the average weight for positive feature was 18.75, the one for neutral was 25, and the average score for negativity was about 56.25, which gives us a negative signal in the daily news, and the signature of that might be the fact that DJIA has dropped in that day, and our binary response variable is 0. Some of the news headlines for this day are "BREAKING: Musharraf to be impeached", "Visitors Suffering from Mental Illnesses Banned from Olympics", and "No Help for Mexico's Kidnapping Surge".

3.4 LSTM Model

We develop an LSTM model for predicting stock market index, DJIA. All pre-processing steps generate input features for LSTM model. We examined the effect of each input feature on the model in experiment section. The final model, inputs, the process of the LSTM architecture, and the output are depicted in Figure (2). The inputs to the model is DJIA time series from section (3.1), and text features obtained in section (3.2) and polarity and subjectivity section (3.3).

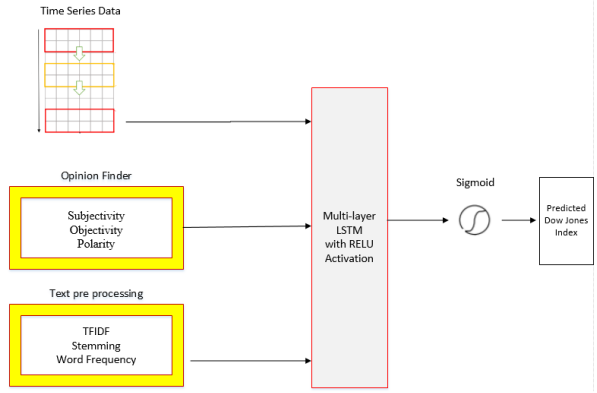


Figure 2: LSTM model

We used a grid search for applying Neural Network on generated features (explained in the previous section). The parameters for grid search are as follows,

- Number of layers: 1-4
- Number of neurons: 32, 64, 128, 256, 512, 1024
- Types of layers: Dense, LSTM
- Activation function
- Target variable: Binary (increase/decrease), Three labels (increase/decrease/same), Regression
- Input Features

The best model obtained from rid search is an LSTM neural network with three layers and hidden units of 512, 256 and 128. The activation function is RELU. The best results is obtained for binary output (increase/decrease).

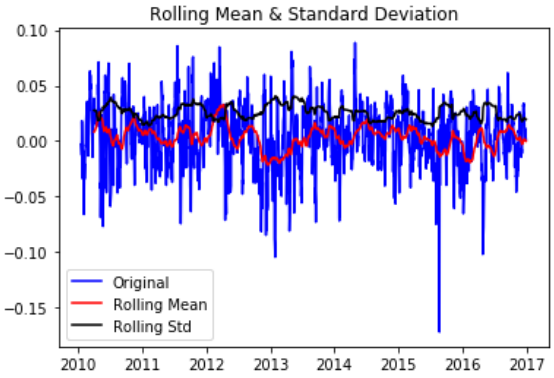
4 Experiment

In this section, we describe all experiments and analysis on the dataset.

4.1 Time Series Forecasting (baseline)

The time series dataset of DJIA is the basic dataset and baseline model for time series forecasting.

An LSTM model is applied on this dataset to forecast stock market, in the baseline case, using just stock market data. This is a challenging problem, as time series values are non-stationary. Using Dicky-Fuller test, the test statistics is -1.5, while critical value is -3.5 in figure (3). We convert the dataset to a stationary dataset by finding the difference of x_t and x_{t-1}. As a result, the dataset becomes stationary by Dicky Fuller test. Here, test statistic become -9 which has lower value than -3.5 as 1% critical value. Therefore, the dataset becomes stationary as it is shown in Figure (3). In addition, we obtained the best Window-size, which is is an important parameter in time series analysis, using grid search. The best windows size which we obtained is three with ROC-AUC score of 0.55. It shows that the time series forecasting without text data on stock market forecasting problem has very low performance.




```
Results of Dickey-Fuller Test:
Test Statistic                -9.461891e+00
p-value                        4.331546e-16
#Lags Used                     1.300000e+01
Number of Observations Used    1.739000e+03
Critical Value (1%)           -3.434116e+00
Critical Value (5%)           -2.863203e+00
Critical Value (10%)          -2.567656e+00
```

Figure 3: Converting Time Series data to a stationary data

4.2 Text Feature Engineering

In this section, we analyze our text pre processing steps. As we describe in section (3.2), features are generated as Linguistic features, TF-idf and Decomposed Embedding vectors.

We used *gradient boosting classifier (XGBoost)* to test these features. Figure 4 shows the top few

features, sorted by their contribution in the classification. For instance, *NumChars5* represents number of characters in the fifth top news headline, and *(W2V75, W2V50, W2V2)* represent decomposed embedding vectors.

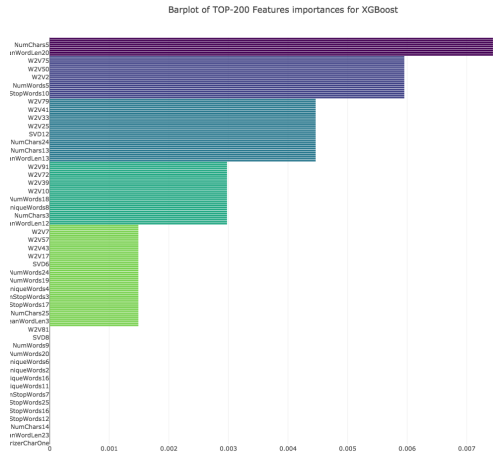


Figure 4: Top features

We also tried *Morphological* Normalizations, namely *Lemmatization* and *Stemming*. The goal of these two transformations is to use a common base for all the different forms of any word in English (e.g., economist, economists, economic, economical, diseconomies, etc. all translate to "economy"). Stemming usually cuts off prefixes or suffixes of the words to get to the root, while Lemmatization uses a dictionary to find the root of the words. Table 4.2 shows the result of applying these two *morphological* normalizations. Obviously, the superior case in terms of better accuracy, would be applying both stemming and lemmatization, at the same time to the text.

Table 1: Morphological Normalization

	F1-Score	**Accuracy**	**AUC**
Raw text	0.6679	0.5159	0.5457
Stem.	0.6703	0.5159	0.5562
Lemm.	0.6547	0.4921	0.3929
Both	**0.6762**	**0.5212**	**0.6211**

The second analysis is on n-gram models. The best obtained model is bi-gram. In the following section, we will explain this method and analyze it.

4.3 N-gram

One of the most popular and basic methods for building language model and text classification is ngram. At first, we built ngram language model from the news, and then using logistic regression, tried to classify them. To eliminate the ineffective frequent occurred word such as 'this', and very rare words, we filtered out the words which occurred in more than 90% of news and also the ones occurred in less than 5 news. We chose these numbers using grid search to obtain the best classification accuracy. Figure 5 shows the histogram of words in bigram model without filtering the frequent words. After removing the frequent words, the number of words reduce significantly (Figure 6).

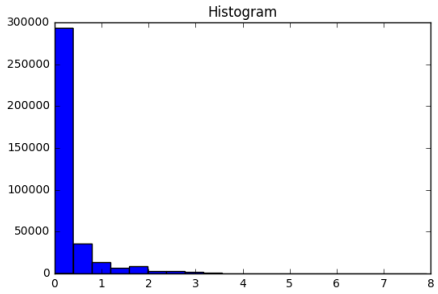


Figure 5: the histogram of words of bigram (without removing frequent words)

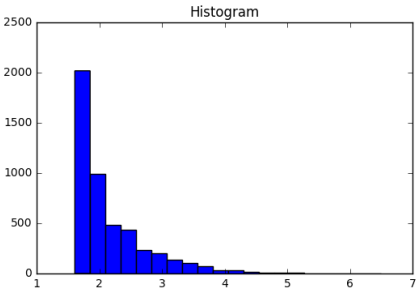


Figure 6: the histogram of words of bigram after removing frequent words

Table 2 shows some of the most effective words, with largest absolute coefficient, for bigram, which was the best language model. The positive words are the words which has positive effect on DJIA index. For example, "haiti earthquake" and "nuclear weapons" had a bad effect on DJIA index, and "population growth" and "air pollution" had positive effect. Some words like "20 percent" might seems to be irrelevant to stock

Table 2: effective words in bigram

	Words
Pos.	says russia,world largest,intelligence agencies, old palestinian,population growth,air pollution, foreign ministry,20 percent, bluefin tuna
Neg.	haiti earthquake,syrian security,1st time, afghan woman,reported killed, sexual abuse, nuclear weapons,boost economy, country world

market, but detected useful by logistic regression model for classification.

4.4 Summary of the results

Overall, the accuracy of the model is not promising for this classification problem. Looking at the previous kaggle kernels, we know that finding a high accuracy for this dataset is hard. Comparing the results of LSTM model with different set of inputs, by looking at the Figure 7, we can conclude that adding all the generated features, would give us slightly better accuracy, after all. We used ROC-AUC score as a measure of performance as labels are imbalanced.

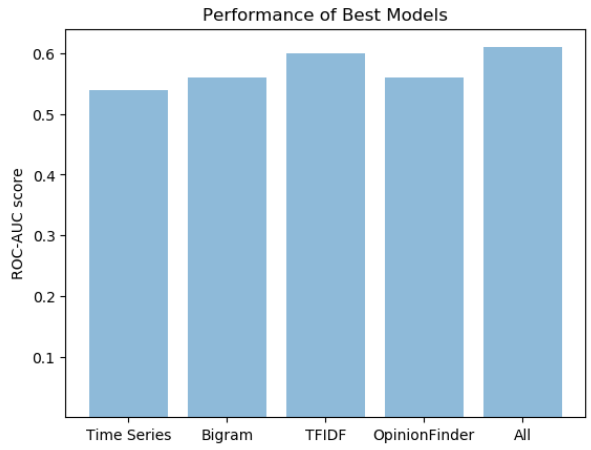


Figure 7: Comparison of Models

5 Conclusions and Future work

As we were expecting, the task of predicting stock market prices, in the near future, and especially in the long-run is very challenging and difficult. Our analysis shows that news headlines, by itself, which reflect the overall status of economy, politics of the day, society, and so on, cannot explain a good portion of the hidden variability in the changes in stock market indexes. It would be best to study the news headlines alongside a set of other factors that most experts believe to be highly influential on the stock market situation. Below are conclusion and possible future works. This dataset is a short dataset of news, including only 25 news. We either need much larger dataset, or combination of different features and methodologies.

Surely a better dataset help to better predict the changes in an index such as DJIA. To be precise, this better dataset could be partitioned with respect to different categories of daily news (e.g., international politics, domestic economy, society, etc.) or with respect to geographic locations (e.g., middle east, Latin America, North America, etc.). It is also a good idea to have longer history of both news headlines and DJIA, so that we can better capture long-term effects such as major recessions. Our data set covers news headlines and DJIA values starting from 2008. However, we know that towards the end of year 2007, there was a major recession in stock market, and due to that millions of people lost their jobs, and the overall trend of stock prices had not changed until after 4-5 years. There were quite similar recessions in the past such as two similar recessions in early 1990's and late 1990's, and one very important and in some sense disastrous one around 1980. Having inclusive data set in this sense, will help us capture this long-term seasonality effect.

We quite liked the idea of *event-driven* news analysis for the stock market prediction proposed in the work by (Ding et al., 2015). We believe better defining these events could potentially help predicting the changes in the stock market. For instance, the events can be defined based on the location in the world, the time-frame they happen during the year, or whether they will have short term or long term effects.

Another direction is to partition the data into different periods during each year. For instance, we know that the dynamics of the market near to Christmas is very different compared to other times during a year. One possible action would be cutting the data related to those 2-3 months, and doing the analysis with our methodology over those specific months, each year. We could do the same for other months of the year, when we do not see a huge change in the tendency of people to buy or sell stocks, mostly based on emotions, rumor, etc., thus creating large economic bubbles in the stock market.

References

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Ijcai*, pages 2327–2333.

Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23.

Helen Susannah Moat, Chester Curme, Adam Avakian, Dror Y Kenett, H Eugene Stanley, and Tobias Preis. 2013. Quantifying wikipedia usage patterns before stock market moves. *Scientific reports*, 3:1801.

Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24):9603–9611.

Nikolaos Pappas, Georgios Katsimpras, and Efstathios Stamatatos. 2013. Distinguishing the popularity between topics: A system for up-to-date opinion retrieval and mining in the web. In *International conference on intelligent text processing and computational linguistics*, pages 197–209. Springer.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics.

Yuzheng Zhai, Arthur Hsu, and Saman K Halgamuge. 2007. Combining news and technical indicators in daily stock price trends prediction. In *International symposium on neural networks*, pages 1087–1096. Springer.