International Journal of Knowledge Based Computer Systems
6 (2), December 2018, 23-30
http://www.publishingindia.com/ijkbcs/

Understanding the Impact of News on Stock Market Trends Using Natural Language Processing and Machine Learning Algorithms

Alka Leekha[1*], Arnav Wadhwa[2], Nikhil Jain[3] and Mehul Wadhwa[4]
[1]Assistant Professor, Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. Email: alka.leekha@bharatividyapeeth.edu
[2]Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. Email: arnavwadhwa14@gmail.com
[3]Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. Email: nikhiljain0106@gmail.com
[4]Department of Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India. Email: mehul9717@gmail.com
*Corresponding Author

Abstract: Short-term trading, specifically day-trading involves a high risk concerning monetary resources. Investing money becomes risky without adequate knowledge and understanding of factors governing the market, some of which include public sentiment, commodity prices, political stability, etc. News is a common way via which people get updates about the latest happenings around the world and hence form opinions about industries, companies, stocks, etc. This affects their trading decisions; substantially impacting their chances of making a profit. Our research focused on building software models that could analyze general news during trading hours and predict the probable stock index closing trend for the end of that day. We used top 25 articles from the Reddit World News Channel and tried to correlate their impact on the DJIA in this study. Using two approaches to the process the text and further deploying machine learning models, we achieved fairly acceptable prediction accuracies.

Keywords: Dow Jones Industrial Average, k-Nearest Neighbors, Logistic regression, Multi-layer perceptron, n-gram model, Reddit World News Channel, Sentiment analysis.

I. Introduction

In popular opinion, investing money into the stock market involves a lot of risk, but provides satisfactory returns if money is put in and withdrawn in an intelligent manner. Making profits require a fair knowledge of the factors governing the market in order to make informed decisions. Certain historic patterns like closing values of stocks- alone don't prove helpful while settling on choices for future since the conduct of a market is conflicting and usually unusual. Multiple factors influence the market and affect a stock price, deciding whether it's value would go up, remain the same, or tumble down, thus introducing dynamicity in its nature. If studied carefully, one can observe that there are indeed certain events and happenings that always trigger a typical corresponding impact on the trajectory of a particular stock's (or industry's) value. With assistance of technological advancements like statistical analysis, machine learning and other analytical techniques (related to data science), software models can be developed that help humans make educated trading choices, which in turn ameliorate the chances of making a profit, rather than letting it be mere odds of luck. Analyzing trends and patterns in the stock market has been a matter of interest among researchers, but there isn't any conclusion derived (yet) which fits universally to every market scenario. If we discuss about public sentiments, we expect it to be significantly shaped by the means of news. These days, almost everyone willing to invest in the market- has access to a multitude of online news articles and utilize it as a source to form opinions about a specific stock, an aggregation, an organization, an industry, or the market in general, which largely impacts their investment decisions.

Many traders are prone to experience losses due to lack of thoughtful decisions. This involves situations where people randomly invest into a stock just because it has an upward trend, without analyzing the company's performance, news on that particular day, important events that might impact the

particular industry, etc. Hence, we believe that it is extremely beneficial to build a 'historic-data-driven' predictive model that can shed light on the probable trends a market can experience, and one crucial factor our model would help analyze, is the real time news via the internet. We believe that such a model can be tremendously helpful for potential investors and would save them from the hassle of analyzing news manually. Data analytics and predictive modeling can be used to develop models that are capable of learning and generalizing the trends in a given set of data, and henceforth predict the responses for unseen data having similar data input formats. Here, prediction means the premonition of immediate future trends of a market, based on current top news. We are focusing only on the top news since it is the news with which majority of population has engaged – in other words, it is the news which has reached the most number of people – leaving impact on a larger segment of the population, that's why it would be on the top.

Our study aimed at developing a software system capable of learning and predicting the impact of everyday top 25 Reddit World News headlines on the Dow Jones Industrial Average (DJIA) closing status. What we really mean to find out is the exact effect of how a person browsing through the Reddit World News Channel might shape their opinion on the companies and industries involved in the DJIA; consequently forming an opinion about investing or not investing money into the index. In order to achieve this, we have used the top news headlines from Reddit World News Channel from a period of 8th June 2008 to 1st July 2016 and the corresponding closing status of the DJIA for each day – as the dataset for building and evaluating ML models. We chose Reddit World News because Reddit provided us with the kind of content we needed for our analysis. Moreover, statistics show that Reddit is the world's 8th most visited website and the United States of America's 4th most viewed website with approximately 1,663 million monthly visitors, which makes it safe to assume that a large number (21 percent) of the world's population consumes the content created by Reddit users, thus leaving an impact on a significant segment of the population.

II. Literature Survey

Various researchers are working on stock price and trend prediction based on social media analytics. In their work, P. Paakkonen and D. Pakkala [1] discussed many use case architectures for building predictive models and applying big data analytics approaches. Till date, various methods have been proposed for quantitatively analyzing the sentiments as well as polarities of text data from sources like twitter, news websites, etc. At first, the mood of a reader about specific company was thought of to be decided by only news / data related to that particular company, but Zhang *et al.* and J. Bollen *et al*. [2, 4] have shown that almost all kinds of news items / tweets contribute to the overall decision of a reader's mood, majorly because in general, people do not only read the financial news, they consume general news as well. In order to evaluate the polarity of news items or tweets, some of the commonly used approaches include the dictionary based approach and the use of semi supervised algorithms for textual data analysis. In the dictionary based approach, the sentiment polarity value of each text item under consideration, is an aggregate of the polarities of each word in that text data, which is found by comparing every word with a labelled dictionary – having words labelled as positive, negative and neutral.

In their research, K. Mizumoto *et al.* [3] talk about semi supervised algorithms, and they say that an initial version of the dictionary is created manually, and then new words are further categorized as being either positive or negative. This is based on the occurrence of new words along with words in the initially created dictionary. As semi supervised learning algorithms might fail to cover every possible combination of words, the dictionary based approach is being used. Interestingly, it can be seen that not only the market affects the moods of the investors, but as J. Bollen *et al.* [4] have analyzed, the mood of individuals in turn also impacts the prices of various stocks. They also concluded that there are only a few companies that directly get affected by the sentiments generated via tweets, i.e. twitter sentiment analysis might be a good factor to be considered in an overall analysis, but not necessarily a very strong one, in all cases. W. Antweilwer and M. Z. Frank [5] discussed that any available information about a company cannot be termed as noise, and in turn, it contributes in one way or another, directly or indirectly, in the prediction of future trends for the company.

R. Ahuja *et al.* [6] collected three months of certain data related to the Bombay Stock Exchange and analyzed the impact of related tweets on that data, achieving a good correlation. N. Lin *et al.* [7] considered the American and Chinese markets, and have also shown that news indeed impacts future trend of stocks. In another work, M. Hagenau *et al.* [8] considered German ad-hoc messages for input to ML algorithms, used the chi-square method for feature selection and further trained a simple linear Support Vector Machine, to obtain a 65% testing accuracy for future trend prediction. J. Gong and S. Son [9] considered feature index variables and built a stock price prediction model using the logistic regression algorithm. They concluded that prediction via the help of appropriately trained logistic regression models usually outperforms other famous methods like the Radial Basis Function Artificial Neural Network (RBFNN) based predictive models.

Coming to the work that has been done related to the concept of Natural Language Processing, there have been many attempts to use features from natural languages like English to ameliorate market trend prediction. Xie *et al.* [10], introduced the concept of tree based representation of the information present in news. In other works, one research made use of text data from twitter; while another research – via the text people posted online, attempted to identify expert investors. Another research studied the impact of news and attempted to figure out the time required to process news in event-driven trading; while

another formulated a method to predict possible potential risk on the basis of financial reports; one other demonstrated that information contained in language based text provides a higher predictability for prices when compared to simple quantitative information.

In 2002, Antonina Kloptchenko *et al.* [11] used data and text mining techniques on financial reports to study patterns about future performance of companies. According to them, annual reports act as a crucial medium for company's communication with stakeholders. In practicality, the approach they proposed can be applied only on reports of a company, and not on websites or achieves. In 2006, Marc-André Mittermayer and Gerhard F. Knolmayer [12] did a survey where to compare 8 text mining models for predicting market response on basis of news. They conclude that none of those techniques had any mechanism of predicting decisions based on fake-news or rumors using natural language processing. In another work, Robert P. Schumaker and Hsinchun Chen [13] proposed a predictive machine learning approach that they called 'AZFinText System' for analysis of financial news articles using various textual representations like the Bag of Words, Noun Phrases, Named Entities. They analyzed over 9000 financial news items and 10 million stock quotes over a period of five weeks. The technique they proposed was efficient and fast, but did not exhibit any historic data driven decision making process.

Vatsal H. Shah with Dr. Mehryar Mohri [14] proposed ML based techniques for prediction where and discussed the application of Support Vector Machines (SVMs), Linear Regression, online learning, and expert weighting in detail, alongside discussion the advantages and disadvantages of all. Nan Li, Desheng Dash Wu in 2009 [15] worked on textual data based sentiment analysis, i.e. emotional polarity computation. Their unique technique revolved around online forum hotspot detection and the use of emotion analysis, text mining for forecasting. Later in 2009, Xiangyu Tang *et al.* [16] presented a stock price forecasting algorithm that combined news data mining with temporal series analysis.

The Efficient Market Hypothesis (EMH) needs to be kept in mind while approaching any market prediction problem, which basically assumes an "ideal" market scenario. It says that security prices gets affected by all available information, and that beating the market is majorly a game of luck, i.e. playing against the odds; rather than a game of skill [17]. In an experiment, the author gathered somewhat around one million tweets that contained certain words potentially impacting stock prices. Twitter sentiment showed 70% accuracy while predicting the actual stock prices and 85% accuracy when the data was included with commodity prices to predict values for individual companies. Data collection is a crucial step that determines the quality of the model being built. Some researchers believe that data from a range of quarter-year to half-a-year months is enough to carry out sentiment analysis and learn a trend.

After data collection, there arises a need to clean and filter the data effectively as per the requirements. Raw and direct data mostly contains information that doesn't contribute to the analysis, and certain text that is not useful at all. Such content needs to dropped and only necessary text needs to be analyzed. Sentiment analysis is not effective without the use of context based dictionaries while analyzing financial news articles. Commonly used dictionaries are focused towards simple English context based text. In order to achieve relevant results in financial context, dictionaries like Loughran and McDonald Financial Dictionary can be used [18]. Also, another problem commonly faced during news based analysis is the fact that that stock trends show a greater reaction to bad news in positive scenarios and relatively a less sensitive reaction to positive news in bad scenarios [19]. A study showed that news has its corresponding highest impact ~20 minutes after release. Attempts have been made to build predictive models based on neural networks, which treat sentiment values of text as inputs, and the exact market value as target variables.

Extreme Learning Machine (ELM) [20], which is a supervised learning algorithm, working on single-hidden layer feed forward networks, has provided better accuracies and prediction speeds; and has also been used for empirical analysis [21]. In their study [22], Chi-Yuan Yeh *et al.* proposed a multi kernel Support Vector Regression (SVR technique) for stock price prediction. It worked by combining multiple kernel matrices, and Lagrange multipliers. Optimal kernel weights were obtained by the learning algorithm by iteratively applying equential minimal optimization and gradient projection, thus giving it the capability to improve system performance by combining the advantages offered by different settings of hyper-parameter tunings. Researchers have also attempted to forecast prices of individual stocks with the help of algorithms like Adaptive Boosting Classifiers (AdaBoost), Support Vector Machine (SVMs), Naïve Bayes classifier, Random Forest, etc. [23].

III. Design and Implementation of Models

A. The Dataset

This dataset is derived from the Reddit World Channel news and DJIA stock price data that a Kaggle user having the name 'Aaron7sun' uploaded. We made use of the everyday top 25 news headlines from Reddit World News Channel from a period of 8th June 2008 to 1st July 2016 and the calculated the corresponding closing status of the DJIA for each day. The dataset contains 27 columns and 1989 rows such that each row represents data regarding one day. Further, the first column represents date; next 25 columns represent the corresponding top 25 news articles for that date (each item in different cell), and one final column depicting a label that represents a fall (represented by 0) OR stagnancy / rise (represented by 1) in the DJIA's closing value for that same day. What this means,

is that for each day- the DJIA Closing value is compared with the closing value of immediate previous day. If the value was higher or same, i.e. the index value rose or stayed the same, a label of 1 was assigned, and if the value dropped, a label of 0 was assigned. This dataset has been summarized below:

TABLE I: DESCRIPTION OF THE DATASET

Column	Description
Date	Date (DD/MM/YYYY) of observation
Top_1	Top user rated news item #1
Top_2	Top user rated news item #2
…	…
Top_24	Top user rated news item #24
Top_25	Top user rated news item #25
Label	(Binary) value depicting stagnancy / rise (1) or fall (0) in the DJIA's Closing Value

B. The Sentiment Analysis Approach

Once the data was collected, sentiment analysis was applied to the news headlines. Regular expressions concept was used for initial data cleaning to remove unwanted information that doesn't contribute to the evaluation of sentiment – e.g. special characters and hyperlinks. Regular expression ("regex") can be defined as a sequence of characters that define a particular pattern. String search algorithms, e.g. the find operation, find and replace operation, etc. can make use of this pattern. This pre-processing provided us with simple alphanumeric text content that was further analyzed to evaluate sentiments.

After the above mentioned pre-processing, sentiment analysis was finally applied using functions from the TextBlob module in python. For each news headline, an appropriate sentiment polarity was evaluated using the sentiment function. A widely used technique for carrying out sentiment analysis is the Lexicon Based Approach [9], based on the prerequisite that a suitable dictionary is available for use. In our approach, we maintained counter variables – 'P' for maintaining the count of positive words, and 'N' for maintaining the count of negative words. During the analysis of text, every word from the text is matched with words in dictionary (a generic English context based dictionary). The dictionary consists of 3 categories (lists), one of positive words, one of neutral, and one of negative words. In case a word from the text is matched in the positive dictionary, the counter 'P' was incremented. In case a word from the text is matched in the negative list, the counter 'N' is incremented. The overall polarity was calculated by using the following formula:

Sentiment Polarity = (P – N) / Total number of words

This provided us with the sentiments of headlines as 25 values, which were added on into the row. Consequently, a table having 1989 rows (just as before – one row for each day), and now having 52 columns was obtained. Every row in this table depicted a single day, and the columns had the data of the 25 news headlines, and in next 25 cells – their simultaneous polarities. In the dataset, the value of polarities of headline's sentiments ranges in [-1, 1]. Further, these were used in a vector of length 25 as a 25-dimensional input feature set for training various classification learners. The corresponding DJIA status was treated as a response (target) variable for each input set. Thus, it can be perceived that the ML models were made capable of taking the sentiments of top 25 headlines of a day at any given time – as input and on its basis, could predict whether the overall DJIA closing value at the end of the day would go up, stay the same, or fall down. To carry out the training and testing of models, a rough 80-20 split was made, i.e., 80% of the data was kept for training and 20% of the data for testing the trained models. This meant that data from the beginning of the dataset up to 31st December 2014 – was used for training and the data from 01st January 2015 till the end of dataset was used for testing the models trained on the training subset.

C. The Language Processing Approach

Here, we made use of the CountVectorizer functionality from Python to carry out our analysis. In this approach, the dataset had to be differently processed in order to be used by CountVectorizer. The pre-processing was done by firstly converting the news articles for each day into a single string – i.e. 25 articles of a day were combined together forming a single string. Then, all the words in the string were converted into lowercase. An example from the corpus is shown below, depicting how the CountVectorizer transforms input data:

- Sample headline picked up from the dataset:

  b "The commander of a Navy air reconnaissance squadron that provides the President and the defense secretary the airborne ability to command the nation's nuclear weapons has been relieved of duty."

- Headline converted to lowercase and then inputted to CountVectorizer:

  b "The commander of a navy air reconnaissance squadron that provides the president and the defense secretary the airborne ability to command the nation's nuclear weapons has been relieved of duty."

- 'Tokens' generated by CountVectorizer that would contribute to the vocabulary**:**

  ['the', 'commander', 'of', 'navy', 'air', 'reconnaissance', 'squadron', 'that', 'provides', 'the', 'president', 'and', 'the', 'defense', 'secretary', 'the', 'airborne', 'ability', 'to', 'command', 'the', 'nation', 'nuclear', 'weapons', 'has', 'been', 'relieved', 'of', 'duty'].

The whole data was processed using CountVectorizer which led to the formation of a vocabulary of 33,875 words (in case of the unigram feature selection approach) and approx. 0.4

million words (in case of the bi-gram word selection approach). This was done by removing all the stop words, i.e. those words which do not contribute any meaning or any useful information to the analysis. These include words like 'a', 'the', 'and', 'an', 'with', etc, which the algorithm is able to do only when it has ample amounts of data. This left out only important words, also called the 'tokens'. Then, the useful features were filtered into a vocabulary & 'counts' were calculated for remaining words.

The 25-headlines-combined-strings, corresponding to each day, i.e. 1989 strings were then compared with the vocabulary which was formed. This was done in order to convert all the combined-strings into their corresponding numerical representations, which was in the form of a 1-D vector for each day. Each 1-D vector contained elements equal in number to the total number of tokens in the vocabulary. Each word inside the 25-headlines-combiend-string was checked with the vocabulary, and only at those places in the vocabulary where the tokens were encountered, a non-zero number was written depicting the 'count' of that particular word in the string under inspection, and in place of those words in the vocabulary where no words from the current string under inspection were present, that position was termed with '0'. This in turn converted all the different day wise 25-headlines-combined-string into their corresponding numerical 2-D array based vector representation in the dataset, and the labels were used as is. Different algorithms were used to fit this dataset and in turn predicting the rise or fall in market giving corresponding accuracies for label 0, label 1 and overall accuracy of the model. The unigram vocabulary based approach and the bigram vocabulary based approach were used to develop the input data for ML models. The results have been summarized in the section below.

IV. Results

The metrics used for evaluation of the models' performance are the training and testing accuracies, precision, recall and f1 scores. In practice, it can be generalized that higher the value of the testing accuracy –the better the model is considered to be.

In case of the Sentiment Analysis approach, after experimenting with predictive models like the Gaussian Naïve Bayes Classifier, Logistic Regression, Support Vector Classifier, Soft Voting of above, and the Random Forest method, we came across results that were only slightly better than random guesses. But while experimenting with the k-Nearest Neighbors and Multi-Layer Perceptron, we achieved results that seemed more rewarding. Going by the logic that in stock market, even if we get successful in predicting the trend 6 times of out 10, it would be better than random guesses, and hence our chances of making money would be better than mere luck.

Table II: Depicting Training Accuracies of the Two Models

Classification Learner	Training Accuracy (%)
K Nearest Neighbors	66.185
Multi-Layer Perceptron	61.690

Table III: Depicting Testing Accuracies of the Two Models

Classification Learner	Testing Accuracy (%)
K Nearest Neighbors	55.52763
Multi-Layer Perceptron	57.83132

A. The Multi-Layer Perceptron – Classifier

A Multi-Layer Perceptron (MLP) is a class of feed forward artificial neural network. A MLP, in theory, consists of at least three layers of nodes. Except for the input node, each node inside the network is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called 'back propagation' for training. It's multiple layers and non-linear activation distinguishes it from a linear perceptron. It has the ability to distinguish data that is not linearly separable. MLP is sometimes colloquially referred to as a "vanilla" neural network, especially when they have a single hidden layer, as in our case.

Parameter Details for hyper-parameter tuning of the MLP model:

- Activation Function: *reLU*
- Value of Alpha: *1e-06*
- Number of Hidden Layers: *1*
- Number of Neurons in the Layer: *4*
- Maximum Iterations: *90*
- Solver: *Limited-memory BFGS*

```
Classification Report for the Training step:

              precision    recall  f1-score   support

           0       0.55      0.60      0.57       670
           1       0.68      0.63      0.65       904

avg / total       0.62      0.62      0.62      1574

Classification Report for the Testing step:

              precision    recall  f1-score   support

           0       0.44      0.55      0.49       154
           1       0.69      0.60      0.64       261

avg / total       0.60      0.58      0.58       415
```

Fig. 1: Depicting the Classification Report in Case of the MLP Classifier Model

B. The k-Nearest Neighbors Classifier

In pattern recognition, the k-Nearest Neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. We used it for classification. The input consists of the k-closest (based on distance, proximity) training examples in feature space. In k-NN classification, the target variable denotes a class membership. A data point is classified by a majority vote of its neighbors, with the point being assigned to the class that is the most common among its k-nearest neighbors (k is a positive integer, typically small).

Parameter Details for tuning the kNN model:

- Number of nearest neighbors, i.e. value of k = *8.*
- Distance formula*: Minkowski,* with value of p = *2, making it a Euclidian Distance.*
- Algorithm: *An automatic selection between kd_tree, ball_tree and brute method.*

```
Classification Report for the Training step:
              precision    recall  f1-score   support

           0       0.74      0.61      0.67       888
           1       0.60      0.73      0.66       703

avg / total       0.68      0.66      0.66      1591

Classification Report for the Testing step:
              precision    recall  f1-score   support

           0       0.62      0.54      0.58       225
           1       0.49      0.57      0.53       173

avg / total       0.56      0.56      0.56       398
```

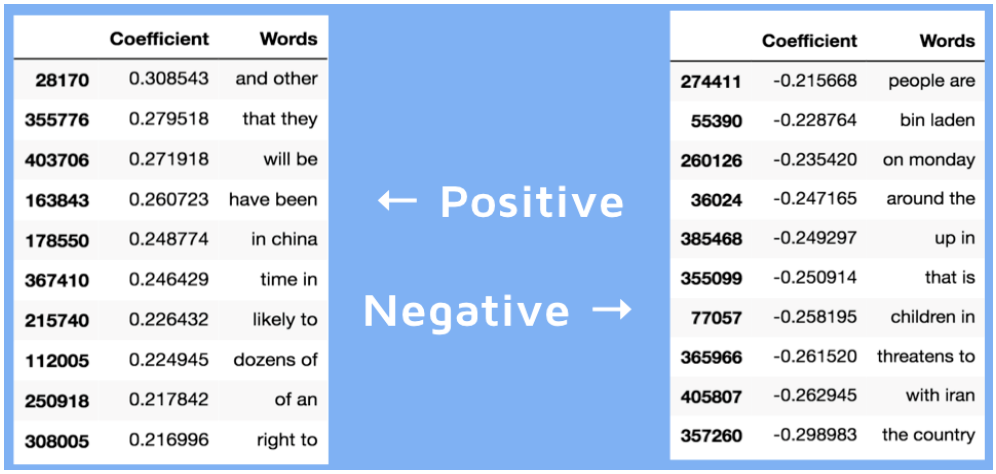
Fig. 2: Depicting the Classification Report in Case of the kNN Classifier Model

Further, in case of the Language Processing Approach, below are the results obtained:

TABLE IV: DEPICTING THE PERFORMANCE OF VARIOUS MODELS ON THE TESTING DATASET

Algorithm (Model)	Prediction Accuracy for Label 0	Prediction Accuracy for Label 1	Overall Accuracy
Logistic Regression (Unigram)	82.22 %	80.80 %	81.48 %
Logistic Regression (Bigram)	86.03 %	83.90 %	84.96 %
k-NN Model (Bigram)	98.40 %	77.5 %	84.90 %
MLP Classifier (Bigram)	88.80 %	83.1 %	85.70 %

From the above table, we can infer that logistic regression for unigram feature based vocabulary gave the lowest accuracy i.e. 81.48% while the MLP classifier for the bigram gave the highest accuracy i.e. 85.70%. MLP classifier constituted of 1 hidden layer and 100 neurons, which gave us the overall best accuracy for this dataset and the algorithm. It also shows us that the bigram based model for logistic regression is better than unigram based for same as bigram gave a comparatively higher accuracy than that of unigram (84.96%). Bigram model means that while forming vocabulary, two words are taken as one unit from the dataset. We also observed via the coefficients for the bi-gram approach, the kind of words that occurred the most during days when market went down and the kind of words that occurred the most during days when market went up.



	Coefficient	Words
28170	0.308543	and other
355776	0.279518	that they
403706	0.271918	will be
163843	0.260723	have been
178550	0.248774	in china
367410	0.246429	time in
215740	0.226432	likely to
112005	0.224945	dozens of
250918	0.217842	of an
308005	0.216996	right to

	Coefficient	Words
274411	-0.215668	people are
55390	-0.228764	bin laden
260126	-0.235420	on monday
36024	-0.247165	around the
385468	-0.249297	up in
355099	-0.250914	that is
77057	-0.258195	children in
365966	-0.261520	threatens to
405807	-0.262945	with iran
357260	-0.298983	the country

Fig. 3: Depicting Most 'Negative' and Most 'Positive' Phrases According to Coefficients

V. Conclusion and Applications

In the Language Processing Approach, the MLP classifier provided the highest testing accuracy among the four (Logistic regression on unigram vocabulary; Logistic regression, k-NN & MLP classifier on bi-gram vocabulary) i.e. 85.70%. Using a model built on this algorithm, the prediction for stock trends based on real-time news articles can be done with a fair accuracy and accordingly, based on predicted trends for the day, short-term traders can make intelligent investment decisions. In case of the Sentiment Analysis Approach, again, the MLP Classifier provided the highest accuracy, i.e. 57.80%. Such software models would be helpful for the investors as they will be able to check before investing their money - whether the market would go up or down resulting in rise / same or fall of stock prices. We also conclude that the DJIA status prediction solely using simply the vocabulary building approach, or the sentiment analysis approach on the everyday top 25 Reddit World News items via classification algorithms, is not highly efficient. One reason for this is – value of stock prices is influenced by a number of other factors as well, like Political Stability, Growth of GDP, Inflation, Liquidity and different interest rate, etc. A complete analysis on basis of just one factor might indeed be helpful, but won't be very fruitful. A predictive model that would take into account all the above-mentioned factors would possibly serve with a much higher accuracy. Companies can use this model to analyze what value they lose / gain from a particular news trend and what kind of incidents triggers investors into investing more money into the market. Apart from this, short-term investors, i.e. people who invest their money in the morning times, when market opens, and plan to make profit by the time market closes, would be prevented from the hassle of going through the entire news items and sources and further analyzing tasks which in fact our model would do for them, hence saving them a lot of time and effort.

References

1. P. Paakkonen, and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Research*, vol. 2, no. 4, pp. 166-186, 2015.
2. X. Zhang, H. Fuehres, and P. A. Gloor, "Predicting stock market indicators through Twitter "I hope it is not as bad as I fear"," *Procedia - Social and Behavioral Sciences*, vol. 26, pp. 55-62, 2011.
3. K. Mizumoto, H. Yanagimoto, and M. Yoshioka, "Sentiment analysis of stock market news with semi-supervised learning," *2012 IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS),* pp. 325-328, Shanghai, 30 May - 01 June 2012.
4. J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1-8, 2011.
5. M. Z. F. W. Antweiler, "Is all that talk just noise? The information content of internet stock message boards," *The Journal of Finance*, vol. 59, no. 3, pp. 1259-1294, 2004.
6. R. Ahuja, H. Rastogi, A. Choudhuri, and B. Garg, "Stock market forecast using sentiment analysis," *2nd International Conference on Computing for Sustainable Global Development (INDIACom),* pp. 1008-1010, March 2015.
7. N. Lin, J. Yuan, W. Xu, L. Wei, and X. Wang, "How web news media impact futures market price linkage?," *Sixth International Conference on Business Intelligence and Financial Engineering (BIFE),* pp. 562-566, November 2013.

8. M. Hagenau, M. Liebmann, M. Hedwig, and D. Neumann, "Automated news reading: Stock price prediction based on financial news using context-specific features," *45th Hawaii International Conference on System Science (HICSS),* pp. 1040-1049, January 2012.
9. J. Gong, and S. Sun, "A new approach of stock price prediction based on logistic regression model," *NISS International Conference on New Trends in Information and Service Science*, pp. 1366-1371, June 2009.
10. B. Xie, D. Wang, and R. J. Passonneau, "Semantic feature representation to capture news impact," *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference*, pp. 231-236, 2014.
11. A. Kloptchenko, T. Eklund, B. Back, J. Karlsson, H. Vanharanta, and A. Visa, "Combining data and text mining techniques for analyzing financial reports," *8th America's Conference on Information Systems*, pp. 20-28, 2002.
12. M.-A. Mittermayer, and G. F. Knolmayer, "Text mining systems for market response to news: A survey," *Working Paper No 184*, Institute of Information Systems, University of Bern, August 2006.
13. R. P. Schumaker, and H. Chen, "Textual analysis of stock market prediction using financial news articles," *12th Americas Conference on Information Systems (AMCIS),* 2006.
14. V. H. Shah, and M. Mohri, "Machine learning techniques for stock prediction," Foundations of Machine Learning, Courant Institute of Mathematical Science, New York University, 2007.
15. N. Li, and D. D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems*, vol. 48, no. 2, pp. 354-368, January 2010.
16. X. Tang, and C. Yang, and J. Zhou, "Stock price forecasting by combining news mining and time series analysis," *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops*, pp. 279-282, 2009.
17. E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383-417, 1970.
18. T. Loughran, and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *The Journal of Finance*, vol. 66, no. 1, pp. 35-65, February 2011.
19. P. Veronesi, "Stock market overreactions to bad news in good times: A rational expectations equilibrium model," *The Review of Financial Studies*, vol. 12, no. 5, pp. 975-1007, 1999.
20. G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, December 2006.
21. X. Li, H. Xie, R. Wang, Y. Cai, J. Cao, F. Wang, H. Min, and X. Deng, "Empirical analysis: Stock market prediction via extreme learning machine," *Neural Computing and Applications*, vol. 27, no. 1, pp. 67-78, January 2016.
22. C.-Y. Yeh, C.-W. Huang, S.-J. Lee, "A multiple-kernel support vector regression approach for stock market price forecasting," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2177-2186, March 2011.
23. A. Wadhwa, N. Jain, M. Wadhwa, and S. Dhall, "Observing the effect of news on stock market using sentiment analysis and machine learning," *Proceedings of the 12th INDIACom, INDIACom-2018, Computing for Sustainable Global Development*, pp. 3754-3757, 2018.