Predictions on Stock Price Change with Data Mining in Top News

Yang Haobo
20656441

Zhang Bowen
20637483

He Junyi
20657093

Wu Zijing
20638346

ABSTRACT

This project contains design, implementation and final evaluation of a model that can be used to predict future stock price based on data analysis from top news headlines. Our project uses the 8 years top 25 news headlines each day from Reddit WorldNews Channel. Machine learning is employed to conduct text classification of data in order to predict the future trend of Dow Jones Industrial Index (DJIA). Our best model using a combined CNN and LSTM model reaches an approximately 58.97% accuracy. Evaluation of our models and idea of a further work are discussed here.

KEYWORDS

DJIA, Stock price prediction, Classification, Machine learning, LSTM, CNN, Logistic Regression, Random Forest

1. Introduction & Relationship

With the advent of information age, it is widely believed that data can bring a lot of information more than just a figure. Recently, the exploration of data is used for practical appl. Stock price prediction is a good application of data mining.

Eugene Fama proposes an efficient market hypothesis (EMH), which pointed out the price of stock can fully reflect all the available information of this asset no matter in the past, present, and future events. We believe that events reported as the top news would result in the change of stock price, especially on significant stock price changing. In order to distinguish whether news of the previous day is related to the change of stock price of next day, we came up the idea that we can build a model using some top news to predict the change of stock price.

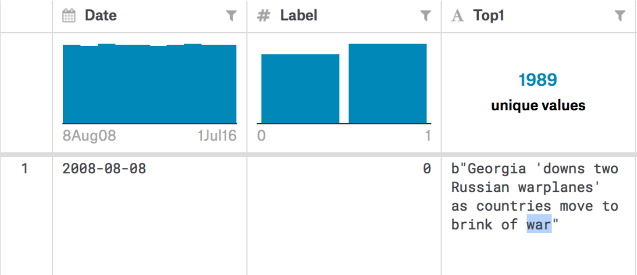
1.1 Data Acquisition & Overview

We obtained our dataset from Kaggle. *(https://www.kaggle.com/aaron7sun/stocknews)*
There are two channels of data provided in our dataset:
One channel of data contains historical news headlines from Reddit WorldNews Channel. They are ranked by reddit users' votes, and only the top 25 headlines are considered for a single date. (Range: 2008-06-08 to 2016-07-01).
Another set of data contains stock data (Dow Jones Industrial Average) (DJIA) containing (Range: 2008-08-08 to 2016-07-01) with open price, highest price, lowest price, close price, volume (number of transactions made) and adjusted close price of a specified date. Directly downloaded from *(https://finance.yahoo.com/quote/%5EDJI/history?p=%5EDJI)*

A combined dataset with 27 columns is also provided. The first column is "Date", the second is "Label", and the following ones are news headlines ranging from "Top1" to "Top25". (There are two labels: "1" when DJIA Adj Close value rose or stayed as the same and "0" when DJIA Adj Close value decreased.)

In order to prove there is relationship between important events and stock price change, our dataset combined news and DJIA value change are briefly overviewed. Some interesting relationships are found in our dataset. If the previous day happened negative events, the stock price tends to fall next day. Also, if the previous day happened positive events that would increase consumer confidence or would lead technological breakthrough, the stock price tends to increase next day.

Take some data for example:
When a war or natural disaster like a tsunami happened, the DJIA value would generally fall next day.



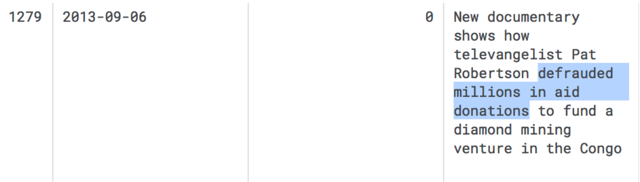
	Date	Label	Top1
1	2008-08-08	0	b"Georgia 'downs two Russian warplanes' as countries move to brink of war"

Figure 1.1: Example 1

698	2011-05-16	0	How one Japanese village defied the tsunami

Figure 1.2: Example 2

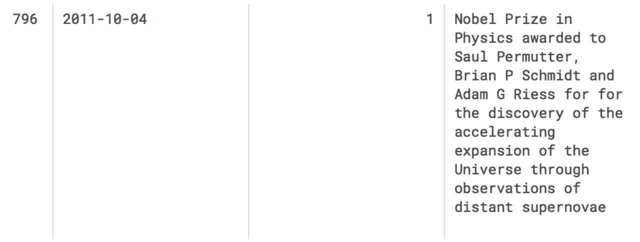
When a major event that significantly influence consumer confidence, the DJIA value would generally fall next day.



1279	2013-09-06	0	New documentary shows how televangelist Pat Robertson defrauded millions in aid donations to fund a diamond mining venture in the Congo

Figure 1.3: Example 3

When a positive event happened, the DJIA value would generally go up next day.



```
796   2011-10-04                                 1   Nobel Prize in
                                                     Physics awarded to
                                                     Saul Permutter,
                                                     Brian P Schmidt and
                                                     Adam G Riess for for
                                                     the discovery of the
                                                     accelerating
                                                     expansion of the
                                                     Universe through
                                                     observations of
                                                     distant supernovae
```

Figure 1.4: Example 4

```
1802  2015-10-05                                 1   Trans-Pacific
                                                     Partnership Trade
                                                     Deal Is Reached
```

Figure 1.5: Example 5

In conclusion, the top 25 news can generally be used to predict some trends of stock prices.

1.2 Data Visualization

Data Visualization is an efficient way to get a comprehensive understanding of the whole pattern. The adjusted close price is chosen to draw a line plot, since the change of adjusted close price is predicted in our models.

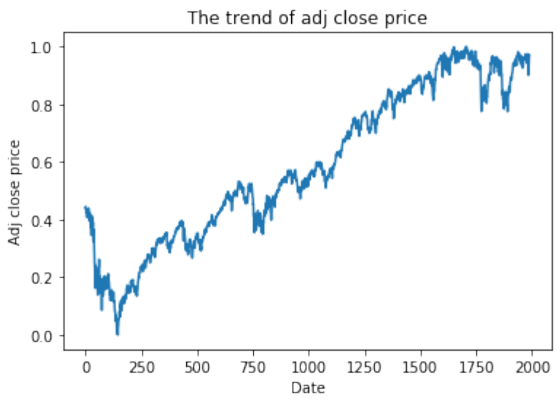


Figure1.6 The trend of adj close price

It is clear that the adjusted close price showed an increasing tendency. Especially from 146th date to 1500th date in our dataset (2019/03/09-2014/07/24), the adjusted close price almost kept monotonously increasing.

1.3 Training and Testing Sets Selection

In order to better train and test the models, the training and testing sets should not be monotonous. Therefore, data on the earliest 85 percent date is chosen as the training set, since data had shown growing and declining during the earliest 85 percent date. And data on the latest 15 percent date is chosen as the testing set in our machine learning model, since data fluctuated greatly and showed irregular change during the last 15 percent date. Also, it is more reasonable that the data in the training set is historical compared to the data in testing set.

2. Logistic Regression

In statistic, logistic regression is used to model the probability of specific events (e.g. win or lose, pass or fail, up or down). Moreover, each input detected in the model will be assigned to two possible results which are 0 or 1, and each output will be assigned to a probability between 0 and 1, and the sum is equal to 1. Logistic regression is generally a two-category problem. For example, if a spam filtering system will be built, x is the characteristic of the mail and predicted y is the category of mails with norm and spam results. A widely method used to separate categorical variables is dividing into positive class and negative class, norm mails is positive (1) and spam mail is negative (0) in this case.

Logistic regression is based on the output of the predicted actual value of linear function $\theta^T x$, looking for a hypothesis function $h_\theta = g(\theta^T x)$ and mapping the actual value to a range between 0 and 1. Moreover, Logistic regression uses the sigmoid function to map the predicted value to the probability value with a range between 0 and 1 to help determine the result. As shown below,

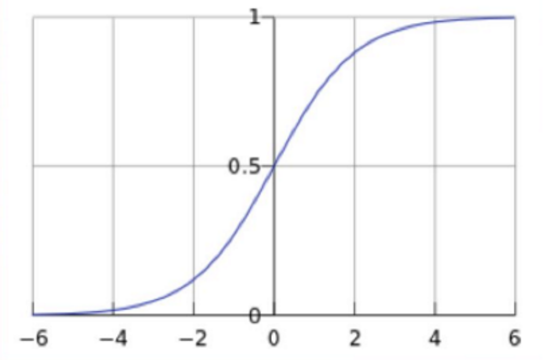


Figure 2.1: The shape of Sigmoid Function

Thus, the mathematical expression of predicted function of logistic regression is $h_\theta = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$, where θ is parameter vector, h_θ is the value of probability of positive class with given x. For making decision, y is predicted to equal to 1 when h_θ is greater than or equal to 0.5, or y is predicted to equal to 0 when h_θ is less than 0.5.

Applicable conditions of Logistic regression model are the followings:

1. Dependent variables are two-category variable or the incidence of an event, and are numerical variables

2. Residuals and dependent variables are subject to normal distribution.

3. A linear relationship between independent variables and the probability of Logistic.

4. Each observing objects are independent.

2.1 Data Processing

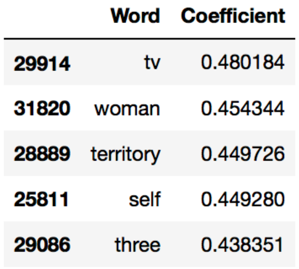
Data cannot be trained directly according to the original dataset, for instance, there are numbers of synonym (airplane and aeroplane) and different forms of words (listen and listening, see and saw, act.) and capitalization of words and so on. Thus, data need to be processed and methods shown below:

1. All uppercase letters are converted into lowercase letters.
2. Delete some dataset which make no sense for machine learning, for example, "b" in title means bolding the title.
3. Delete non-letter symbols, such as @, $.
4. Synonym and different forms of words are converted to same word.

In the next step, the processed news headline is disassembled into a sequence of words and counted the frequency of occurrence of each word. Generally speaking, the frequency of occurrence of each word always has positive relationship with the importance of the word. However, this phenomenon may result from large numbers of definite article existed ("a", "an", and "the"), which is meaningless. The frequency of occurrence of each word too low may be caused by the word is too rare. In order to avoid the problem, filter out words that are included in more than 95% of news headlines, and filter out words that are included in news headlines below 1%. Furthermore, CountVectorizer in Sklearn function would be used to convert the original dataset to TF-IDF （Term Frequency–Inverse Document Frequency） matrix.

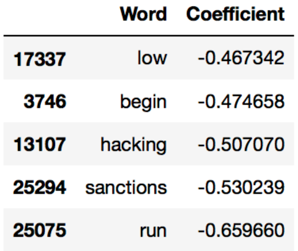
2.2 Modeling1-Logistic Regression with One-gram Dataset

Firstly, the logistic regression model is built with a one-gram dataset, which means that the frequency of each word is an input. The coefficients indicate the relationship between the specific words and DJIA. From the results , the five key words that have the highest positive contributions on the DJIA are listed below, which are ‘tv’, ‘woman’, ‘territory’, ‘self’ and ‘three’. the five key words that have the highest negative contributions on the DJIA are ‘run’, ‘sanctions’, ‘hacking’, ‘begin’ and ‘low’ (Figure2.2 & 2.3).



	Word	Coefficient
29914	tv	0.480184
31820	woman	0.454344
28889	territory	0.449726
25811	self	0.449280
29086	three	0.438351

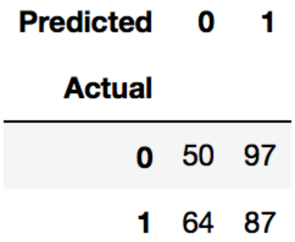
Figure 2.2: Top5 coefficient _1



	Word	Coefficient
17337	low	-0.467342
3746	begin	-0.474658
13107	hacking	-0.507070
25294	sanctions	-0.530239
25075	run	-0.659660

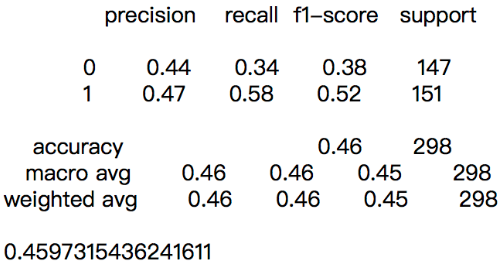
Figure 2.3 Tail5 coefficient_1

As shown in Figure 2.4, the accuracy of this model is only 45.97%, which is significantly lower than Random speculation (50%). Moreover, there are other alternative accuracy measures can be used, which shown in Figure 2.5.



Predicted	0	1
Actual		
0	50	97
1	64	87

Figure 2.4 Confusion Matrix_1



```
              precision    recall  f1-score   support

           0       0.44      0.34      0.38       147
           1       0.47      0.58      0.52       151

    accuracy                           0.46       298
   macro avg       0.46      0.46      0.45       298
weighted avg       0.46      0.46      0.45       298

0.4597315436241611
```

Figure 2.5 Performance Measure_1

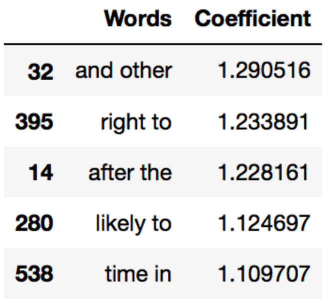
Deriving from Figure2.5, when the interest class is "1" (i.e. the trend of DJIA would increase), the precision is 0.4597, which means that of the 184 (97+87) cases that the model predicts it will rise and 87 truly happen. Secondly, the recall is 0.58, which means that only 87 cases are retrieved by the model in the 151(64+57) cases that happened. For F-measure, it is the weighted harmonic average of Precision and Recall and used for evaluating the quality of the model. The mathematic function is

$$F_\beta = \frac{(\beta^2 + 1) * PR}{\beta^2 P + R}$$

, where β is a parameter, P is precision and R is a recall. F1-Measure is the parameter equal to 1. The closer the F-value is to 1, the better the model. However, F-score is only 0.52 for the 1-gram logistic regression model.

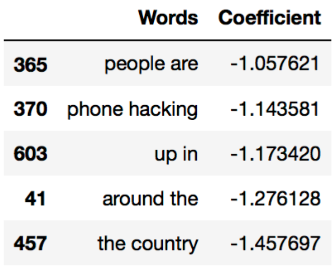
2.3 Modeling2-Logistic Regression with Two-gram Dataset

Due to the low accuracy of the first logistic regression, a two-gram dataset was constructed to build a new model. By constructing two words into a group, the numbers of input would increase dramatically. The figure2.6 and 2.7 shows that he five key words that have the highest positive contributions on the DJIA are ‘and other’, ‘right to’, ‘after the’, ‘likely to’ and ‘time in’, and the negative contributions on DJIA are ‘the country’, ‘around the’, ‘up in’, ‘phone hacking’ and ‘people are’.



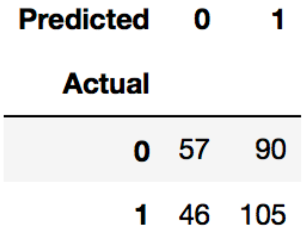
	Words	Coefficient
32	and other	1.290516
395	right to	1.233891
14	after the	1.228161
280	likely to	1.124697
538	time in	1.109707

Figure 2.6 Top5 coefficient _2



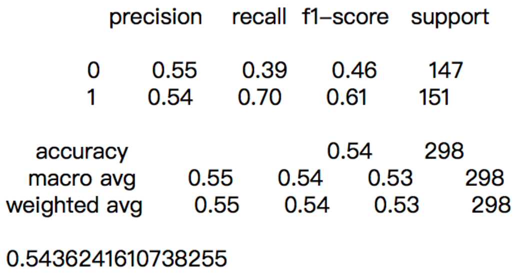
	Words	Coefficient
365	people are	-1.057621
370	phone hacking	-1.143581
603	up in	-1.173420
41	around the	-1.276128
457	the country	-1.457697

Figure 2.7 Tail5 coefficient_2



Predicted	0	1
Actual		
0	57	90
1	46	105

Figure 2.8 Confusion Matrix_2



```
              precision    recall  f1-score   support

           0       0.55      0.39      0.46       147
           1       0.54      0.70      0.61       151

    accuracy                           0.54       298
   macro avg       0.55      0.54      0.53       298
weighted avg       0.55      0.54      0.53       298

0.5436241610738255
```

Figure 2.9: Performance Measure_1

Deriving from Figure2.8 and 2.9, it is clearly showing that the accuracy of two-gram logistic regression is obviously enhanced from 0.4597 to 0.5436. In addition, when the interest class is ‘1”, the precision is 0.54, meaning 105 cases will truly rise with the total 195 (90+105) cases predicted by the model. The recall of 0.70 shows that 105 cases retrieved by the model with 151 (105+46) cases that actually happened. Lastly, F-score is now 0.61 and closer to 1 than the previous one, indicating the 2-gram logistic regression is better than 1-gram.

3. Prediction Using Random Forest

3.1 Introduction of Random Forest

Random forest is a kind of classifier by taking advantage of many Classification Decision Trees which are used to train samples from population.

Random forest is a holistic learning method for classification, regression and other tasks. This method operates on the number of individual trees by constructing a large number of decision trees during training and outputting classes as class patterns or mean prediction. Random decision forest corrects the habit that decision tree adapts too much to its training set.

Random forests differ only from general schemes in that they use an improved tree learning algorithm to select a random subset of features at each candidate segmentation in the learning process. This process is sometimes referred to as "feature bagging." The reason for this is the correlation of the trees in a generic bootstrap sample: if one or several features are very strong predictors of the response variable (target output), then these features will be selected in many b-trees, thus making them Become related. The influence of bagging and random subspace projection on the accuracy under different conditions is analyzed.

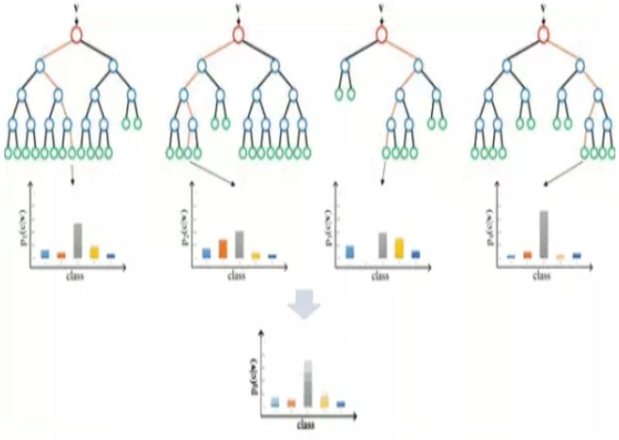


Figure 3.1 Flowchart of random forest method

Above is the simplified flowchart of random forest method. The first step is illustrated by plotting four graphs which are four

Classification Decision trees, it is the first procedure of fostering random forest model.

After randomly choosing samples and attributes from the whole population and then input them into the random forest model, many trees could be generated. The next stage is input the test set into these trees and each tree is able to make a prediction. Finally, the final decision could be made by combining each prediction with a function.

A random forest is a kind of algorithm using a decision tree (CART) model. For example, 200 thousand observation in the total population and 20 variables. Random forests attempt to construct various CART models with different samples and different initial variables. For example, it will construct a CART model using 10 thousand observation of random samples and 5 randomly selected initial variables. Repeat the process many times and subsequently get a final prediction of the observations each time. The final prediction is a function of each prediction. In some cases, final prediction could be the average of each individual observation.

3.2 Advantages of Random Forest

Random forest has some benefits compared to some other methods, which are shown below:

1. It can process thousands of input variables without variable deletion.
2. It gives an estimate of which variables are important in the classification.
3. As forest fostered progresses, it produces internally unbiased estimates of generalization errors.
4. It has an efficient way to estimate missing data and maintain accuracy when most data is lost.
5. It provides method of balancing errors in unbalanced data sets in a cluster.
6. The generated forest can be saved for future use in other data.
7. It provides information about the relationship between variables and classifications.
8. It calculates the relationship between case pairs that can be used for clustering, positioning outliers, or (by scaling) giving interesting data views.
9. These characteristics can be extended to untagged data, resulting in unsupervised clustering, data view and outlier detection.
10. It provides an experimental method for detecting variable interactions.

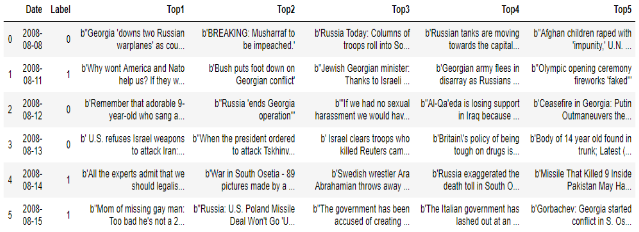
To sum up, this algorithm plays an important role in the application of decision tree, moreover, it is also able to make an accurate prediction in many cases.

3.3 How Random Forest Works?

There are many trees for classification in the forest and if we want to classify an input sample, we need to input sample into every individual tree and let these individual trees to make a classification respectively, then sum up the results and the class with the highest frequency will be the final prediction of the class of sample.

3.4 Our Random Forest Model in DJIA Prediction

The first stage is to choose a dataset that we use to build the model and divided it into training set and test set, here we assign the earliest 85% of the whole data to training set in date order(from early to latest), and assign the rest 15% to test set.



	Date	Label	Top1	Top2	Top3	Top4	Top5
0	2008-08-08	0	b"Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b"Afghan children raped with 'impunity,' U.N. ...
1	2008-08-11	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b"Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b"Olympic opening ceremony fireworks 'faked'"
2	2008-08-12	0	b'Remember that adorable 9-year-old who sang a...	b"Russia 'ends Georgia operation'"	b'"If we had no sexual harassment we would hav...	b"Al-Qa'eda is losing support in Iraq because ...	b'Ceasefire in Georgia: Putin Outmaneuvers the...
3	2008-08-13	0	b' U.S. refuses Israel weapons to attack Iran:...	b"When the president ordered to attack Tskhinv...	b' Israel clears troops who killed Reuters cam...	b'Britain\'s policy of being tough on drugs is...	b'Body of 14 year old found in trunk; Latest (...
4	2008-08-14	1	b'All the experts admit that we should legalis...	b'War in South Osetia - 89 pictures made by a ...	b'Swedish wrestler Ara Abrahamian throws away ...	b'Russia exaggerated the death toll in South O...	b'Missile That Killed 9 Inside Pakistan May Ha...
5	2008-08-15	1	b"Mom of missing gay man: Too bad he's not a 2...	b"Russia: U.S. Poland Missile Deal Won't Go 'U...	b"The government has been accused of creating ...	b'The Italian government has lashed out at an ...	b'Gorbachev: Georgia started conflict in S. Os...

Figure 3.2 A part of the whole dataset

The table above is a proportion of the whole data set, we can see that the first columns from left is the date from August 2008 to July 2016, the second column is "label" where cells in this column get two different values "1", "0", here "1" represents increase or unchanged in DJIA close value, and "0" represents decrease. The flowing columns are Top25 historical news headlines in each date.

Then we need to process the raw documents and transform them into a matrix of TF-IDF(Term frequency-Inverse Doc Frequency) features, finally it will be transformed into TF-IDF-weighted document-term matrix. The TF-IDF works as weight factor in information index.

Below is the formula of calculating TF-IDF:

$$tf(i,j) = \frac{frequency\ of\ term\ i\ in\ j\ document}{number\ of\ all\ terms\ in\ j\ document}$$

*It denotes the frequency of the term i in document j

$$idf(i) = \log\left(\frac{D}{number\ of\ documents\ including\ term\ i}\right)$$

*The less the number of documents including term i, the higher the score of the above paramitor

$$tf\_idf(i,j) = tf(i,j) \times idf(i)$$

*If measure the importance of term i in document j

$$tf\_idf(i) = idf(i) \times \sum_{j}^{document\ number\ n} tf(i,j)$$

*High value of the parameter means term i play an important role in classification.

Pre-processing of historical news headlines is finished by using the function "TfidfVectorizer" in python. Here use the hyper parameter "min_df" and "max_df" to ignore some terms. Min_df is set to be 0.01 which means the least frequent 1% of terms is ignored; max_df is set to be 0.99 denoting the most frequent 1% terms is ignored, for example, some words such as " a, the, is, are" need to be ignored, otherwise, these words can make noises in training trees in the random forest. “ngtam_range=(1,1)” , we first process by using single word, this may ignore the combination of different words which may arise some different meaning. Therefore, we change the hyper parameter to “(2,2)” later to make a comparison.

The next step is training the Random Forest model with the function “RandomForestClassifier()”, and make the prediction of test set.

To evaluate the accuracy, precision, recall, f1-score, we calculate these indexes by using flowing formulas:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

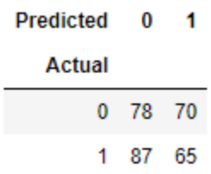
$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN}$$

$$F_\beta = \frac{(\beta^2 + 1) * PR}{\beta^2 P + R}$$

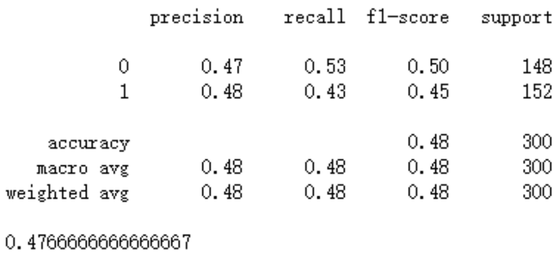
	Class 1 predicted	Class 2 predicted
Class 1 actual	TP(true positive)	FN(false positives)
Class 2 actual	FP(false negatives)	TN(true negetives)

When we use one-gram dataset (use single word), the accuracy is 0.47666667:



```
Predicted  0   1
Actual
        0  78  70
        1  87  65
```

Figure 3.3 Confusion matrix_1



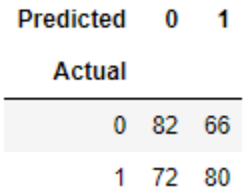
```
              precision    recall  f1-score   support

           0       0.47      0.53      0.50       148
           1       0.48      0.43      0.45       152

    accuracy                           0.48       300
   macro avg       0.48      0.48      0.48       300
weighted avg       0.48      0.48      0.48       300

 0.4766666666666667
```

Figure 3.4 Performance Measure_1

As we can see from figure 3.3 and 3.4, the accuracy is so low that even failed to reach 50%.

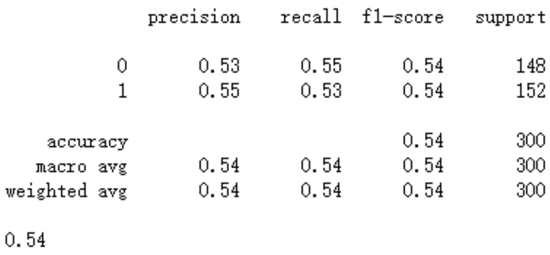
Based on the result, we try to optimize the hyper parameter. Therefore, we change the “ngtam_range=(1,1)” to “ngtam_range(2,2)”, which means combine two words to build the TF-IDF-weighted document-term matrix.

As a result, the corresponding performance is shown below:



```
Predicted  0   1
Actual
        0  82  66
        1  72  80
```

Figure 3.5 Confusion matrix_2



```
              precision    recall  f1-score   support

           0       0.53      0.55      0.54       148
           1       0.55      0.53      0.54       152

    accuracy                           0.54       300
   macro avg       0.54      0.54      0.54       300
weighted avg       0.54      0.54      0.54       300

 0.54
```

Figure 3.6 Performance Measure_2

From the above two graphs, we can see that accuracy has been increased to 0.54, so the combination of two words is proved to be more reasonable.

4. Prediction Using Combination of Convolutional and Recurrent Neural Network

Sentiment analysis of short texts is difficult because there is limited contextual information they contain. The mixed model of CNN and RNN takes advantage of the coarse-grained local features in CNN and long-distance dependencies learned by RNN.

4.1 Data Cleaning and Pre-processing

For each day, the news data contains top 25 headlines. There are 1940 valid days and total 48500 headlines. Each headline has different words. First, we need to do some data cleaning and pre-processing. All characters will be converted to lower case, and unwanted characters and stop words will be removed. Remaining part of the words will be converted to the pre-trained vector based on the GloVe's pre-trained word vectors.

GloVe is an unsupervised learning algorithm for obtaining vector representations for words.[1]

If the words are not found in the vocabulary, the random embedding will be created for them. Hence, all words will be represented by the number vectors. Here we will use the Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB) pre-trained word vectors, which means every word will be embedded into a 300-dimension number vector. The reason why to use pre-trained embedding instead of random embedding is that the Euclidean distance (or cosine similarity) between two word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words, which may improve the performance of sentiment analysis. For example, in our vocabulary, word ‘b’ will be represented by a 300-dementional vector:

[-5.11709988e-01, -1.06810004e-01, -4.06890005e-01,
……
-1.95570007e-01, 9.43770036e-02, 1.42859995e-01]

Another important pre-processing is that due to the limitation of CPU performance and time. We should balance the training time and performance. Here we will limit the max daily length to 200 words. Since every input length should be the same, missing words from the vocabulary will be replaced by <UNK>, and sentence shorter than the fixed length will be added <PAD> until the length reaches the correct length. Since the range of output dataset is large and the variance is also high, all price changes will be normalized to zero to one.

Also, in this model, the output will be a normalized number between 0 and 1 instead of binary output. Becase we want our model to learn different news with different weights to improve the accuracy. The inputed daily change will also be normalized to 0 to 1.

4.2 Model Construction

After the pre-processing, we need to decide the model used to learn the data. Our idea is to use neural network since our input and output may have deep connections that are not so obvious to be observed. [2] Our model is inspired by the paper Combination of Convolutional and Recurrent neural network for sentiment analysis of short texts. [3] The model combines the advantages of CNN and LSTM networks. The detailed structure of our model will be shown in next several parts.

CNN Layer

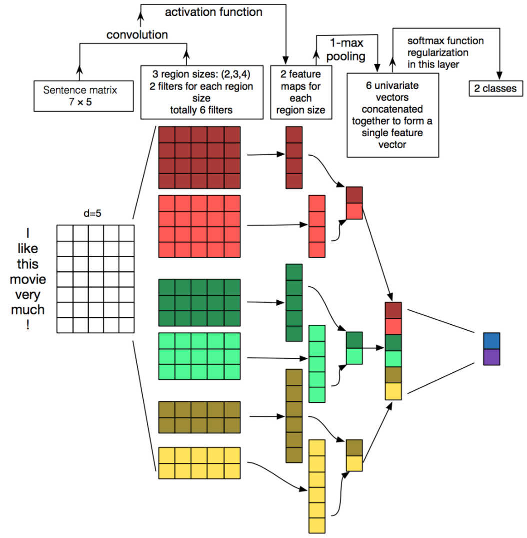

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 4.2 An example of illustration of a Convolutional Neural Network (CNN) architecture for sentence classification.
Image reference:
http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

At the beginning of the network, words will be embedded to the integer vector through an embedding layer. Hence, the shape of one input to CNN will be (max_length_of_one_day_news, embedding dimension). The idea of using convolution neural networks for sentence classification comes from paper Convolutional Neural Networks for Sentence Classification [2]. The paper reports that a simple convolutional neural network with hyper parameter tuning achieves excellent results on multiple benchmarks.

LSTM Layer

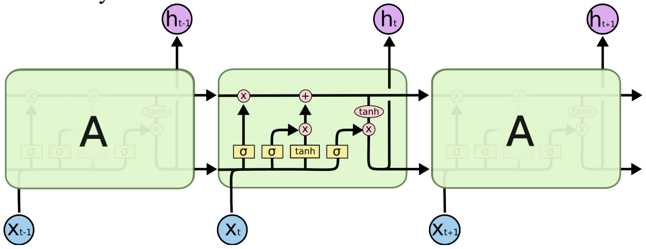


Figure 4.3 Sample LSTM network
Image reference: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM networks are already widely used in sentence classification. In paper Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts [3], a new method is introduced to take advantage of the coarse-grained local features
generated by CNN and long-distance dependencies learned via RNN for sentiment analysis of short texts.

Dense Layer

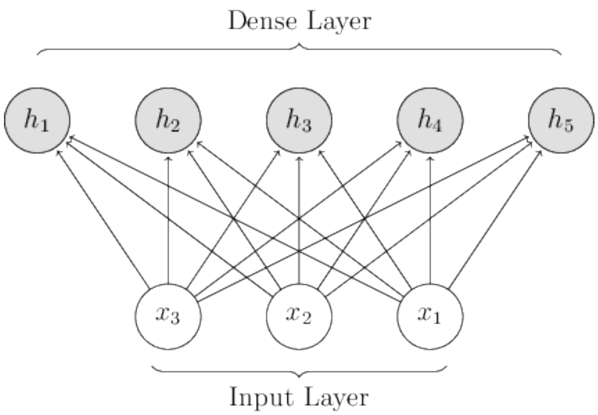


Figure 4.4 An example of single hidden dense layer

LSTM layer will be connected to one dense layer which more numbers of units compared to the LSTM output units with 'relu' activation function. And finally, the dense layer will be connected to an output layer which has only one output unit. Different from binary output, our model's output is a specific number between 0 and 1, which will be un-normalized at the end. But based on the un-normalized result, we can still get a binary result. Lager than 0 indicates the price will increase, less than 0 indicates the price will decrease.

Dropout Layer

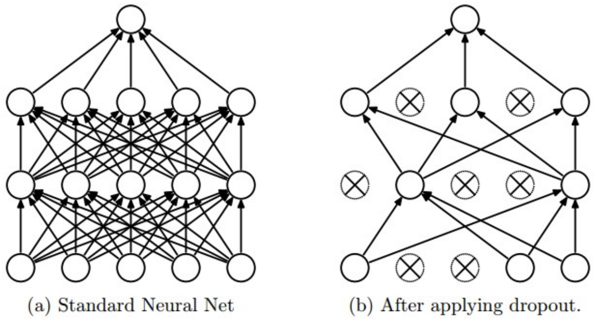


Figure 4.5 Different between networks with and without dropouts

Image reference: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/dropout_layer.html

Dropout layers are used to prevent over fitting in neural networks. In our model, there will be many dropout layers because the huge dimension of embedding word vectors.

Embedding layer

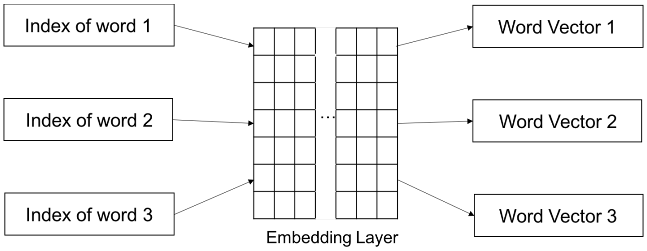


Figure 4.6 Word embedding in an embedding layer

The use of embedding layer is to convert the word index (input) to the target word vector. The embedding transformation is mainly processed by a word embedding matrix generated by the pre-trained GloVe word vectors. Thus, we can convert the nature language processing to a normal machine learning task.

4.3 Overview of Entire Model

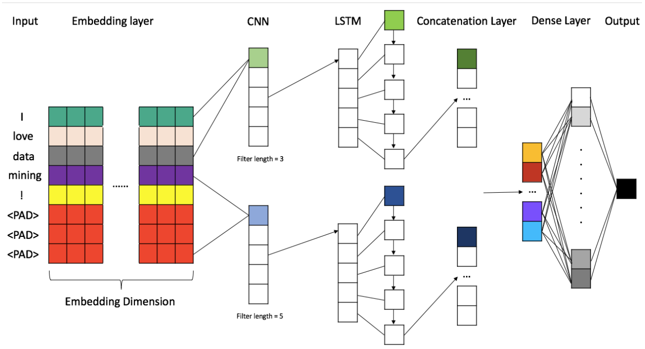
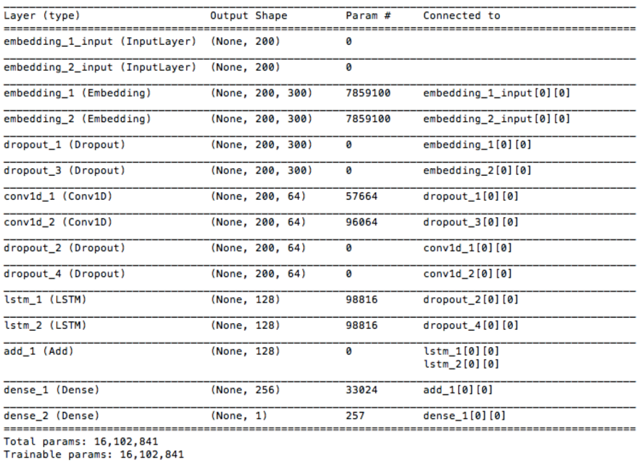


Figure 4.6 Detailed structure of model. The dropout layers are hidden for convenience.

Our model combines the advantages of CNN and LSTM. The first layer will be embedding layer used to convert word integer index to corresponding pre-trained word vector. After that, two different CNN network will be used. The number of filters are the same, but the kernel size is different, 3 and 5 for each. After each convolutional neural network, a long short term memory network will be connected. Wang (2016) showed that this kind of neural network can improve the performance in sentiment analysis of short texts. Then, a concatenation layer will do add function to the two output of different LSTM, which use the information collected by two different CNN of different kernel size. Finally, the dense layer will take the output of long short term memory network as input and calculate the weights and give the result, which is between 0 and 1 due to the previous normalization in data processing.



```
____________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================================
embedding_1_input (InputLayer)  (None, 200)          0
____________________________________________________________________________________________________
embedding_2_input (InputLayer)  (None, 200)          0
____________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 200, 300)     7859100     embedding_1_input[0][0]
____________________________________________________________________________________________________
embedding_2 (Embedding)         (None, 200, 300)     7859100     embedding_2_input[0][0]
____________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 200, 300)     0           embedding_1[0][0]
____________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 200, 300)     0           embedding_2[0][0]
____________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 200, 64)      57664       dropout_1[0][0]
____________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 200, 64)      96064       dropout_3[0][0]
____________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 200, 64)      0           conv1d_1[0][0]
____________________________________________________________________________________________________
dropout_4 (Dropout)             (None, 200, 64)      0           conv1d_2[0][0]
____________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 128)          98816       dropout_2[0][0]
____________________________________________________________________________________________________
lstm_2 (LSTM)                   (None, 128)          98816       dropout_4[0][0]
____________________________________________________________________________________________________
add_1 (Add)                     (None, 128)          0           lstm_1[0][0]
                                                                 lstm_2[0][0]
____________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          33024       add_1[0][0]
____________________________________________________________________________________________________
dense_2 (Dense)                 (None, 1)            257         dense_1[0][0]
====================================================================================================
Total params: 16,102,841
Trainable params: 16,102,841
```

Figure 4.7 The detailed layer and output information

The figure 7 shows the information of each layer including the output shape and the number of parameters. Because of the high dimensions of word vectors, the number of total parameters will be large compared to normal neural networks.

After the hyper parameter optimization and model adjustment, our final model can achieve 58.97% accuracy on testing set in predicting the increase or decrease in stock price. There is an interesting finding that the experiments go against the conventional knowledge of the more layers the better. Add some more max pooling layers, dense layers, CNN or LSTM layers will reduce the accuracy to around 53% to 56%.

4.4 Result

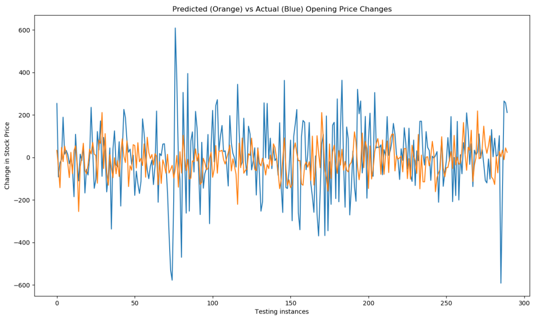


Figure 4.8 Predicted change in stock price on testing data

As shown in figure 8, the orange line shows our predicted change in stock price. The predict accuracy is 58.97%. There are some good features that our model has.

First, our model can detect many significant changes in stock price, like increasing to decreasing in a very short time. In our model, different from predictions using time series data, each prediction by our model is stand-alone, which means it cannot learn any historical data. Hence, our model can correctly capture some important news that can influence stock price immediately, which cannot be predicted by time series predictions.

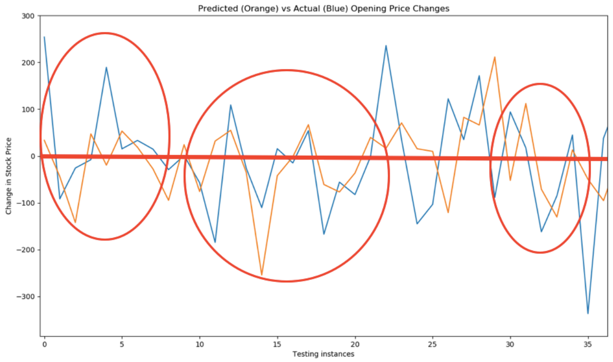


Figure 4.9 Our model successfully captures the significant change in stock price

Secondly, as shown in figure 10, given the stock price of first day, 2015.1.1, to be 0, our predictions fit the actual data well in 100 days, which is not a short time. This result is totally out of our expectation, because each prediction is calculated only on the news headlines stand-alone, without any history stock price data. The prediction tendency is almost the same as the actual data, which means even combine all the predicted results, our model is still reliable though the predictions are only expected to match the price change of next day. It also indicates that even the accuracy is not so high, it is still reliable and reasonable. For a clear comparison, we generate five different stock price change lines using random steps in normal distribution of all previous data (mean is 3.2657 and stand division is 141) in figure 11 along with the actual data.

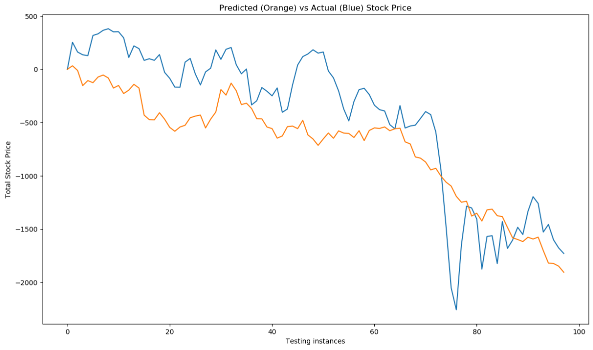


Figure 4.10 Predicted stock price in 100 days from 2015.5.11

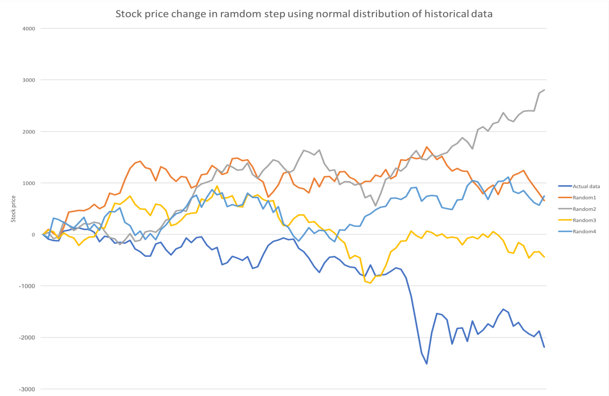


Figure 4.11 Stock price change in random step using normal distribution of historical data

4.5 Drawbacks

This model cannot learn from historical data and are not sensitive to increasing tendency. The drawbacks can be obviously viewed in the predicted 35 days data from 2015.10.01.

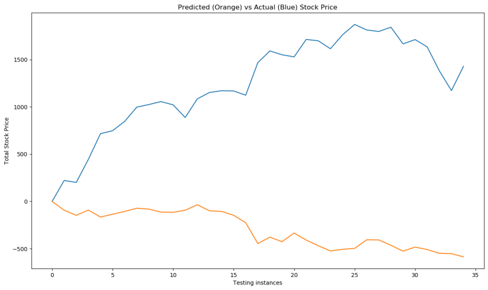


Figure 4.12 Predicted stock price in 35 days from 2015.5.11

5. Time Series Analysis Using Historical Data

5.1 Time Series Analysis vs. Text Analysis on Stock Price Prediction

The past value of stock price can reflect the future trend of stock price. Therefore, time series analysis is widely used for stock price prediction. However, we believe that there is some limitation on time series analysis. Tseng and Chen (2005) proved that It cannot solve some fundamental problems like time delay, and it is hard for time series model to detect sudden change. [4]

Also, we believe that some sudden change of the stock price may be reflected in some important events and can be detected by text analysis. Therefore, a comparison between time series analysis and text analysis is made.

5.2 Building the LSTM Model & Results

A simple LSTM neural network is built, concretely let Xt denote the adjusted close price in date t. Then:

$$Xt=f(Xt-1, \Theta)$$

Here Xt is the the adjusted close price at time step t, Xt−1 denotes number of passengers at the previous time step, and Θ refers to all the other model parameters, including LSTM hyperparameters. Our LSTM model consists of one LSTM layer of 4 blocks, one dense layer to produce a single output and window size of 50. MSE is used as loss function in our model with epochs=20. The LSTM model predict the future adjusted close price based on 50 historical data. And we get a prediction line that relatively fits to the true value.

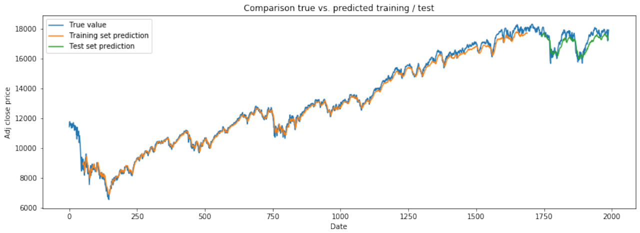


Figure 5.1 Comparsion true vs. predicted training/test

Then we make a comparison between the change of stock price and get the below graph.

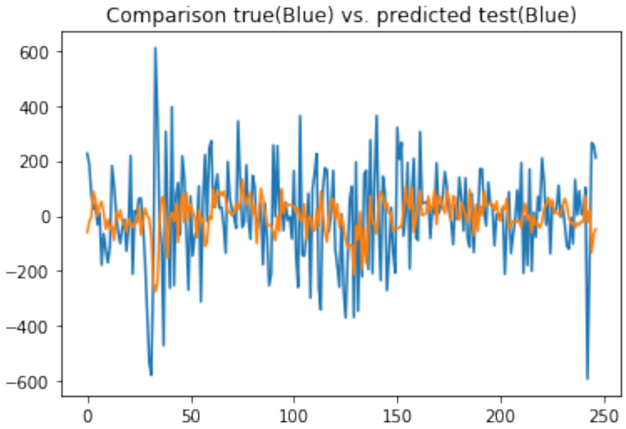
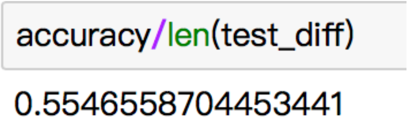


Figure 5.2 Comparison true vs. predicted test on difference

From Figure5.2, the accuracy of predicting the tendency of stock price seems not to be high. Then we compute the accuracy and got:



```
accuracy/len(test_diff)

0.5546558704453441
```

Figure 5.3 Accuracy of trend prediction

The accuracy is approximately 0.55, which is even lower than text analysis. Therefore, we conclude that our time series model can precisely predicts and detects continuous changes but rarely detect the sudden changes.

6. Future Work

Our CNN and LSTM combined model and time series model both have some drawbacks. It is hard for the CNN and LSTM combined model to detect increasing tendency and predict accurately during a long period, which may can be solved by the time series model. Therefore, combining these two models can significantly increase the accuracy of prediction. We were considering developing a model combined historical data and news data, but we do not finish it due to limited time. We believe that the accuracy can be increased to approximately 70% in the improved model.

Conclusion

In this project, we want to predict the future change of stock price using top news headlines. It is a challenging work because there seems not so much connections between our inputs and outputs. We use different machine learning models to finish the tasks and achieve good performance on the prediction accuracies. We think there should be a deep connection between news and the stock price since the financial market will be influenced by many human factors, which will be presented on the top news. Our works also agree that predictions of stock price change based on news are meaningful and useful. Most importantly, during the whole project, all of our teammates learn a lot in machine learning methods, model construction, data processing and coding.

ACKNOWLEDGMENTS

We would like to express our appreciation to all those who provided us the possibility to complete this report expecially our teammates. A special gratitude we give to professor James Kwok, who contributes in providing machine learning knowledge and inspiration.

CODE

Our project code and data can be viewed on:
https://github.com/Dotafterfootball/MFIT5004

REFERENCES

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[2] Kim, Yoon (2014). *Convolutional Neural Networks for Sentence Classification*. arXiv:1408.5882.

[3] Wang, Jiang & Luo. (2016). *Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts*. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers , Page 2428–2437, the COLING 2016 Organizing Committee.

[4] Tseng, Vincent & Chen, Yen-Lo. (2005). *An Effective Approach for Mining Time-Series Gene Expression Profile*. 10.1007/11539827_20.