Text Driven Stock Market Prediction Based on Machine Learning Methods

Team Member

WEI, HU	**54206763**
Yidi, LIU	**54765801**
Dongliang, SHENG	**54918176**
Qiping, WANG	**54486639**
Kai, YANG	**54878893**

Executive Summary

In this study, we attempt to investigate how to extract useful information from news headlines and utilize it to predict financial market. We propose four models namely ARIMA, SVM, RNN and LSTM to predict firm-level stock price trend (whether the specific stock price will be up or down in the next day) using information from stock price or news headlines. Specifically, we first employ a time series model, ARIMA model, to predict price trend of seven technology firms, which are Apple, Amazon, Facebook, Google, Microsoft, IBM, and Tesla, using only stock price data. Meanwhile, we also adopt three machine learning methods namely SVM, RNN and LSTM to conduct the price trend prediction task for the same seven firms, using only news headlines. According to our experimental results, we uncover four major findings: firstly, three machine learning methods which only use news headlines data all outperform the conventional method ARIMA model which only uses stock price data, indicating that news headlines can help predict financial market; secondly, the sentiment-based method SVM outperforms the other three none-sentiment-based models in predicating stock price trend, indicating that sentiment is one factor that can be used to predict price trend; thirdly, different news show different predictive power, indicating that only news headlines that are relevant to the firm can makes significant difference in financial market trend prediction ; fourthly, time lags exist in stock price trend forecasting when using news headlines and different firms show different days of time lags, indicating that different investor groups exist in different firms and special news released during that time period.

1 Introduction

Stock price prediction is always a hot topic of interest both in industry and academic area. A great many methods haven been adopted by practitioners and researchers to enhance stock price prediction accuracy. Among them, sentiment analysis is one of the most well adopted approaches in financial market prediction. Abundant studies have demonstrated the significance value of investor sentiment in stock price forecasting (Baker and Wurgler, 2006). Therefore, investor sentiment extracted from news can be helpful for stock market prediction(TETLOCK, 2007), and sentiment from news headlines a popular and well-built method(Strapparava and Mihalcea, 2008) with less noise. In this sense, sentiment extracted from news headlines is promising to generate great research potential in predicting stock prices.

However, one of the existing challenges is how to extract useful information from news headlines, and utilize such information to predict financial market. Another challenge is that though many researchers attempt to capture the trend of stock price movements through various propose models and information sources, yet the prediction accuracy varies across different models and different information sources.

Our project attempts to address the aforementioned business issues. We propose four models namely ARIMA, SVM, RNN and LSTM to predict firm-level stock price trend (whether the specific stock price will be up or down in the next day) using information from stock price or news headlines. Specifically, we first employ a time series model, ARIMA model, to predict price trend of seven technology firms, which are Apple, Amazon, Facebook, Google, Microsoft, IBM, and Tesla, using only stock price data. Meanwhile, we also adopt three machine learning methods namely SVM, RNN and LSTM to conduct the price trend prediction task for the same seven firms, using only news headlines.

According to our experimental results, we uncover four major findings: firstly, three machine learning methods which only use news headlines data all outperform the conventional method ARIMA model which only uses stock price data, indicating that news headlines can help predict financial market; secondly, the sentiment-based method SVM outperforms the other three none-sentiment-based models in predicating stock price trend, indicating that sentiment is one factor that can be used to predict price trend; thirdly, different news show different predictive power, indicating that only news headlines that are relevant to the firm can makes significant difference in financial market trend prediction; fourthly, time lags exist in stock price trend forecasting when using news headlines and different firms show different days of time lags, indicating that different investor groups exist in different firms and special news released during that time period.

The rest of the report proceeds as follows. Section 2 involves a description of data collection process, including seven high-tech companies' stock price data and news headlines. Section 3 describes our proposed methods, including one conventional model and three machine learning models. Section 4 interprets the experimental results for four specific approaches. Section 5 discusses our research findings, provides a concluding remark and highlights the directions of our future research work.

2 Data Description

This section describes the data collection process, including stock data and news headline data. For news headline data, we obtain our news headlines data from an open dataset on Kaggle. The dataset contains 422,419 news headlines of 11,237 websites, ranging from March 10th 2014 to August 26th 2014. For stock data, we download seven high technology firms' stock price data from yahoo finance. Accordingly, the time period for stock data is also from March 10th 2014 to August 26th 2014. Generally, the stock price data is mainly for ARIMA model, and the news headlines data is for three machine

learning methods. Figure 2-1. illustrates relevant data source, data type and the usage of specific data.

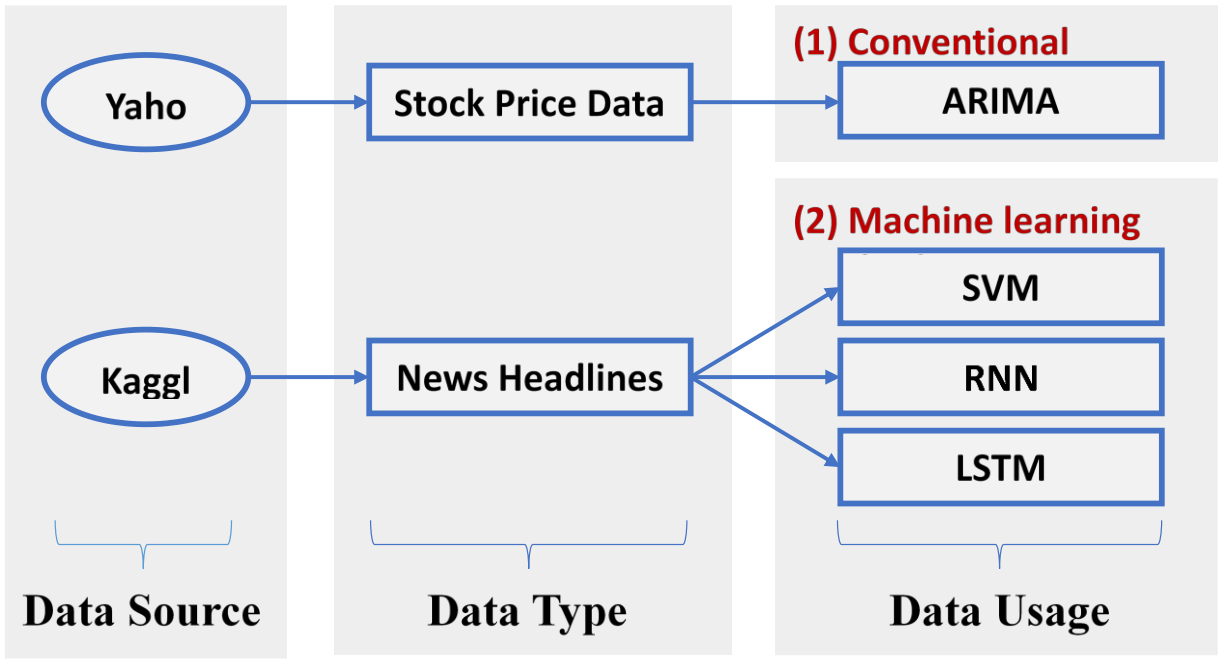

Figure 2-1 Data usage framework

2.1 Stock Data

This study proposes four specific methods and utilize them to predict stock market. Therefore, collecting stock price data is the premise. We choose seven well known high technology companies in the United State, namely Apple, Amazon, Facebook, Google, Microsoft, IBM and Tesla. The reason we choose these seven firms as our target company is that compared to other small and non-famous companies, they are all famous companies listed in U.S., which are more possibly to attract media attention and release more news every day. We download these seven companies' stock price data from Yahoo Finance, which provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content. The time period is from March 10th 2014 to August 26th 2014. Thereinto, our evaluation time period is from March 10th 2014 to June 30th 2014, and our prediction time period is from July 1st 2014 to August 26th 2014. Figure 2.1-1 is the stock price

fluctuations for these seven companies over seven months.

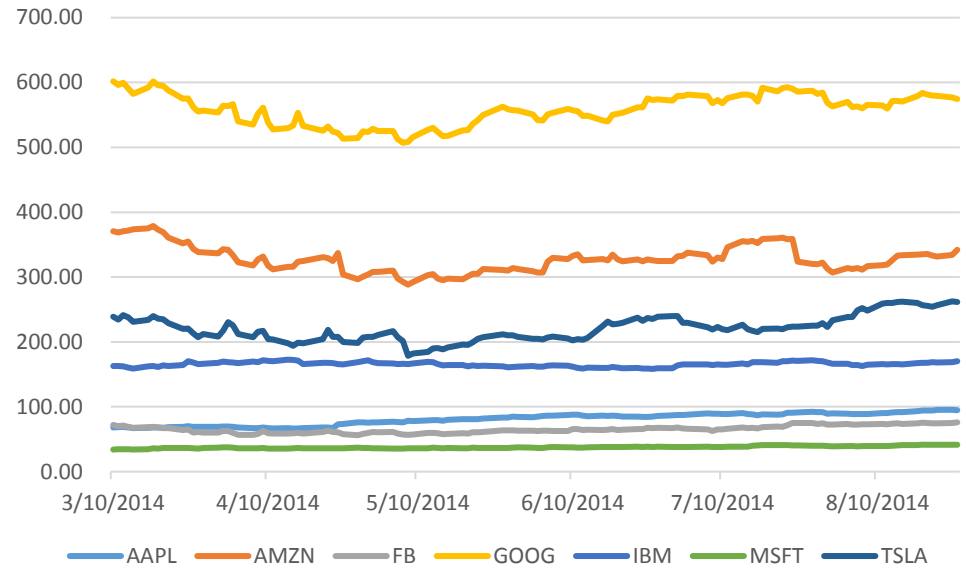

Figure 2.1-1 Stock price fluctuations

2.2 News Headlines Data

For news headlines data, we found an open dataset on Kaggle. The dataset contains 422,419 news items scraped from 11,237 websites. It includes eight columns: ID (the numeric ID of the article), TITLE (the headline of the article), URL (the URL of the article), PUBLISHER (the publisher of the article), CATEGORY (the category of the news item), STORY (alphanumeric ID of the news story that the article discusses), HOSTNAME (hostname where the article was posted), and TIMESTAMP (approximate timestamp of the article's publication, given in Unix time). We include four categories of news topics, i.e. business, technology, entertainment and health.

Figure 2.2-1 is a pie chart for each category of news. Figure 2.2-2 Shows news clusters for each category. Our seven target firms are all belong to the technology sector. In addition, we further separate our news headlines data into three type: (firm) relevant news, all technical news, and (firm) relevant technical news. Take Apple Inc. as an example, (Apple) relevant news contains all four categories of

news regarding to Apple, all technical news means all news regarding to the technology category(not limited to Apple), and (Apple) relevant technical news refers to technical news regarding to Apple.

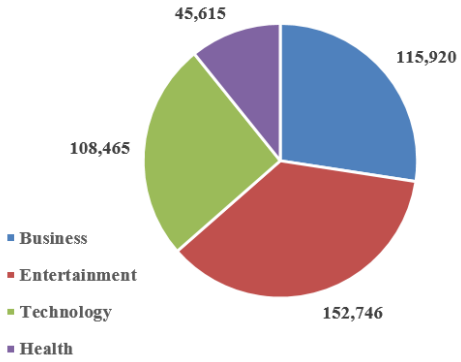

Figure 2.2-1 News category

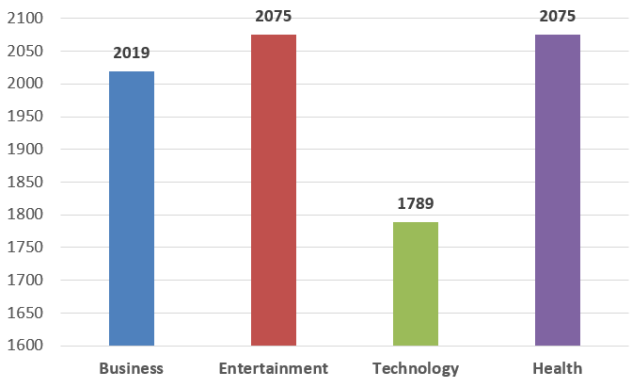

Figure 2.2-2 News clusters

3 Method

3.1 Price-Based: ARIMA

To make comparisons with the case of utilizing news headlines, the first kind of predictions was based on previous stock price data only, without any other information. As time series data, ARIMA models can be used to make more accurate predictions.

For each company, 10 different ARIMA models were used, which are ARIMA(0, 1, 1), ARIMA(1, 1, 0), ARIMA(1, 1, 1), ARIMA(1, 1, 2), ARIMA(2, 1, 1), ARIMA(2, 1, 2), ARIMA(1, 2, 1), ARIMA(1, 2, 2), ARIMA(2, 2, 1), ARIMA(2, 2, 2).

In order to keep the consistency of other machine learning models, the evaluation criteria is based on the price trend of a certain company's stock, instead of the specific predicted value of the stock price. The evaluation of ARIMA models are based on the AIC and BIC, and the prediction accuracy.

Similarly, to compare with other machine learning methods, the same test dataset was used. Here, the predictions of stock trend for all different methods were all based on the stock price in July. For ARIMA model, the time series data was from March 10th 2014 to June 30th 2014. The prediction time

period was the whole July of 2014 (July 1st 2014 to 26th 2014), which were in all 40 transaction days.

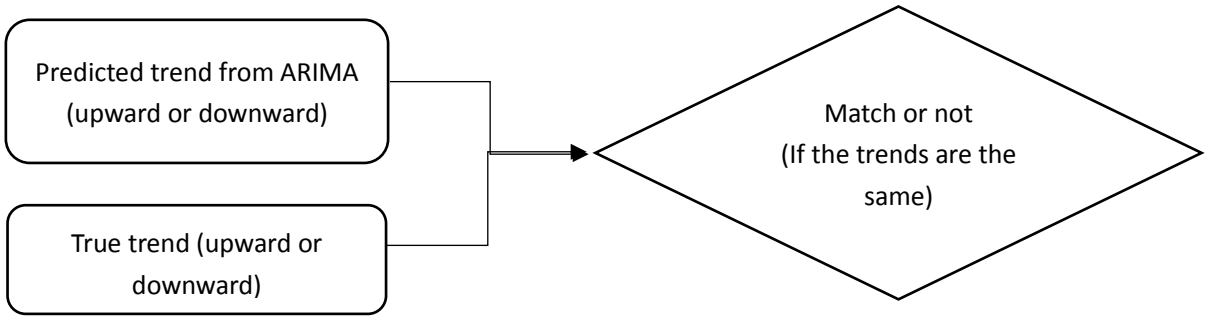

Figure 3.1-1 The prediction accuracy from ARIMA

Some conclusions can be made from the analysis of ARIMA models. First of all, for each company, the performance of ARIMA models were not satisfactory. Most models' prediction accuracy was less than 50%, which means relying only on previous time series data of stock price, the prediction of trend is not accurate and unstable. Secondly, best ARIMA model for each company showed time lag. Most of the results showed autocorrelations of previous stock price.

Although ARIMA is a good viable method to make predictions of time series data, the accuracy is not very satisfactory, even ten different ARIMA models were used for each company to make predictions of stock price trend.

3.2 Sentiment-Based: SVM

As a reflection of a thought, an opinion, or an attitude, the influence of sentiment has been researched a lot. In behavioral finance, studies have shown that social mood can influence people's investment behavior. For example, researcher proposed theory of social mood to clarify the influence of sentiment (Nofsinger, 2005). Recent researches also showed that sentiment derived from news can show helpfulness for the prediction of financial market, such as stock price and trend (Tetlock et al, 2008).

Utilization of news is a viable method to investigate the influence of sentiment. News can reflect

the market response and professionals' opinions. To save time and extract sentiment more accurately, news headlines are a good source. News headlines can not only reflect the main story of a news report, but also provides the basic sentiment tendency. Sentiment extraction from news headlines is a popular and well-built method (Strapparava and Mihalcea, 2008) with less noise.

The news headlines are categorized into three types. The first type includes all the relevant news. The second type is made up of all the technology news, and the last type is the relevant techonology news, which is the intersection of the first two types.

The measurement of sentiments extracted from news headlines is a specific value, which is from -1 to 1. For each day, the sentiment is calculated as the average of all the sentiments extracted from news headlines on the same day. To find the relationship between the stock price trend and the sentiment, those two dimensions were plotted. Figure 3.2-1 shows the time series of stock price and sentiment for company Apple.

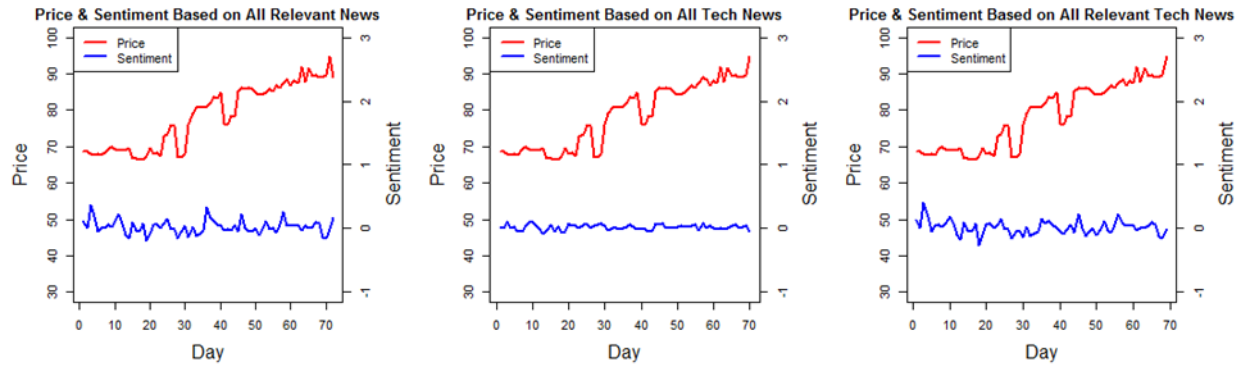

Figure 3.2-1 Price &and sentiment of Apple

Based on those data and plots, the relationship between the sentiment and stock price trend is not easy to conclude. To further analyze the influence of sentiment on stock price trend, the correlation between sentiment and stock price trend for each company was made. Table 3.2-1(Appendix) shows the correlation between the sentiment of each type of news and the stock price trend.

From this correlation table, several conclusions can be made. First of all, similar with ARIMA model, the influence of sentiment showed time lag as well. It was found that most of the largest correlation did not happen on the same day, but showed lag of one day, or two days, or three days. For example, for company IBM, the correlation of all relevant news's sentiment and stock price trend for time lag one day, two days and three days were -0.008, -0.046 and 0.057 respectively. However, the correlation jumped to 0.301, which was quite large. This also happened on the other companies. Hence, it is found that the response of stock price does not necessarily happen instantly. There exists a time lag in financial market prediction when using the sentiment from news headlines.

Secondly, the effect of positive sentiment may not result in upward trend of stock price. From the correlation table, it is found many of the correlations are negative. Even though there exists significant correlation between the sentiment from news headlines and stock price trend, the effect can be both positive and negative according to different companies and different time lags. For example, the correlation between sentiment and stock price trend for company Apple is negative for two-day time lag, but for IBM, the correlation is positive for three-day time lag.

It is also found Google's response is much more instant. The largest correlation happens on the same day with the news. Possible reason may be that the news is much more important, such as big event and announcement. Additionally, it is also decided by the nature of Google, which is the largest search engine company.

Figure 3.2-2 shows the overall process of stock market prediction model. First of all, we have the source news headline data. We select the following three types of news headlines and construct three dataset for each company: all relevant news contains all the news headlines that include the company name or company ticker; tech relevant news contains all the news headlines in the "technology"

category that include the company name or company ticker; all tech news contains all the news headlines in the “technology” category.

Next, we conduct text data pro-processing to structure and clean our data. The source data is unstructured, and thereby our first step should tokenize the documents, and represent each document with a fixed length vector using the Bag-Of-Word model. Although in the Bag-Of-Word model, the sequence of words in a document will be ignored, it seems that the sequence information is not important in our task since our objective is to apply lexicon-based tools to analyze the sentiment of documents. Since most of headlines only contain one sentence, we treat each headline as a document in our experiment. During the tokenization process, we remove those stop words since they contribute less to the documents’ sentiment. Instead, we retain punctuation, since some punctuation like “!” or “?” can express the sentiment to some extent. We also conduct stemming process to transfer words into their normalized form. The reason is that there exists less sentiment information in the tense of a headline, and if we retain the normalized form of each word, we will get a very sparse document-word matrix eventually. In this experiment, we only use unigrams and do not use N-grams, the reason is that the sentiment analysis tools we use require the input to be the unigrams.

From previous steps, each documents have been converted into fixed-length vectors. Then, in the sentiment analysis step, we adopt a popular Python toolkit in NLP, called NLTK, to analyze sentiment of headlines. NLTK provides API for over 50 corpora and lexical resources, e.g., WordNet, and sentiment lexicon. NLTK provide powerful tools for sentiment analysis. It identifies sentiment of a document through calculating the average score of the sentiment words appeared in the document. The input of the NLTK sentiment analysis tool is a vector representing a document, while the output of it is a sentiment score. Actually, this is just like a dimension reduction process where the high dimension

representation is converted into only one dimension. The sentiment scores range from -1 to 1. Score 0 means neutral emotion, and -1 means negative emotion while 1 means positive emotion on the contrary. Although there are many other sentiment lexicon especially those financial lexicon, e.g. financial times lexicon, we only choose NLTK sentiment lexicon, because it has been widely used in many previous literature especially in the stock market prediction context (Yu et al., 2013). The sentiment scores are regarded as features for the machine learning methods. As we presented in the previous section, there exists a time lag effect on the correlation between headlines' sentiment and companies' stock price trend, which means that the sentiment of news will not affect the stock market immediately, but instead, generate influence several days after. For different companies, they have different time lag. Besides, the correlation pattern between sentiment and stock price trend is not always positive. In other words, positive sentiment news may not necessary lead to good stock market performance.

In sentiment analysis, the simple way to predict the price trend using sentiment is to predict upward trend if the sentiment is positive while downward trend if the sentiment is negative otherwise. However, the results seems bad because the correlation between sentiment and stock price trend is complicate and good news may not necessary result in good stock market performance.

To tackle this question, we utilize a machine learning method, support vector machine (SVM), to automatically capture the correlation patterns. In the field of machine learning, SVM is a supervised learning method that is usually used for pattern recognition, classification and regression analysis. The main idea of SVM can be summarized as the following two points. First, it can address the linearly inseparable cases. A non-linear mapping algorithm is used to transform the low-dimensional input space which is linearly inseparable into high-dimensional space which is linearly separable, and thereby linear algorithm can be applied in the high-dimensional space to conduct the linear analysis.

Second, SVM constructs the optimal hyperplane in the feature space based on the structural risk minimization theory, which maximize the model, and the expectation of the entire sample space satisfies a certain upper bound with a certain probability. In this project, we utilize a Python package "sklearn", which is an open source machine learning library including various classification, regression and clustering algorithms, to implement the SVM model. Since the input of SVM is only one dimension, it is proper to utilize the simple linear kernel. We set the penalty parameter for the error term as 1, which obeys the common practices. The dataset is divided into training set and testing set. Headlines published before 2014-07-01 belong to the training set and those after 2014-07-01 belong to the testing set. In this experiment, we do not use validation set. The validation set is applied to adjust the parameter in machine learning models and to prevent over-fitted. In this project, since SVM has been widely applied in many sentiment analysis scenario, there exists some common practice about the parameter setting, and therefore we do not need to adjust the parameter using the validation set. Using the training set, we train the SVM model. Then, we input the testing set into the trained model, and get the prediction results.

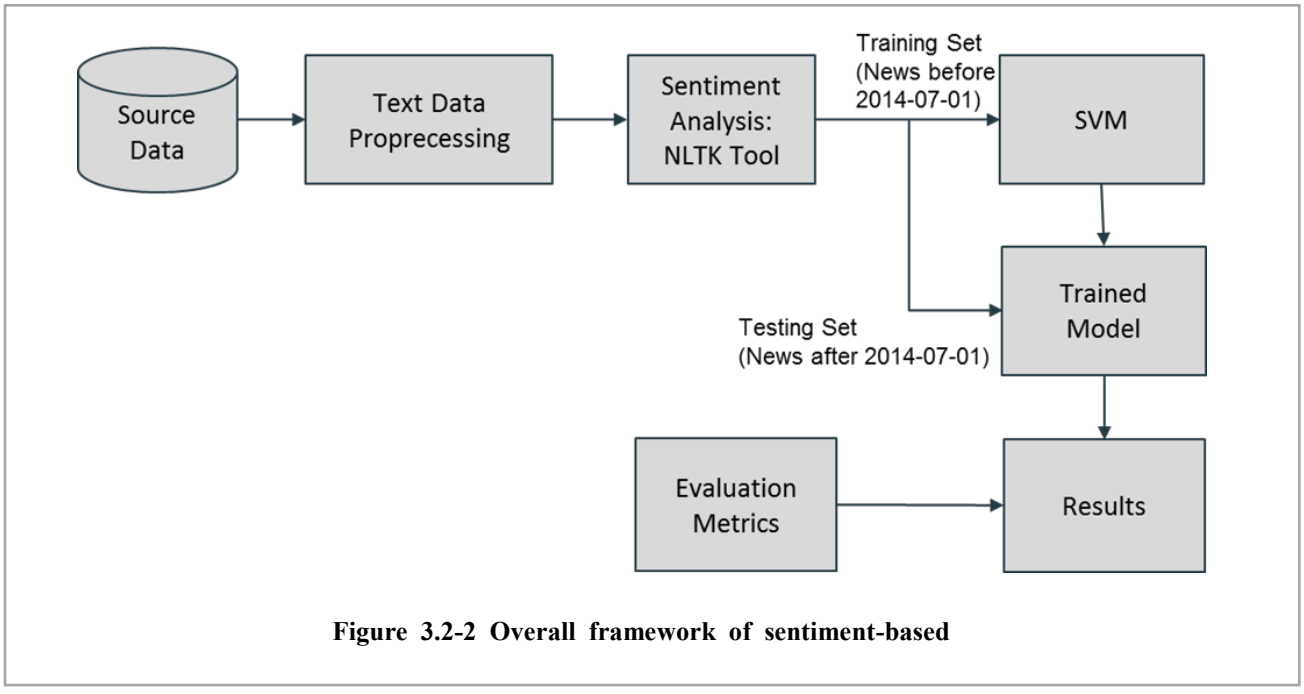

Figure 3.2-2 Overall framework of sentiment-based

In this study, we choose the accuracy as our evaluation metrics. The reason is that we are interested in both true-positive and true-negative values. The accuracy is calculated as follows:

$$\text{ACC} = \frac{TN + TP}{TN + TP + FN + FP}$$

The results of this model will be shown in section 4.2.

3.3 Non-Sentiment-Based: RNN

In this section, we compose a non-sentiment method to address the market prediction problem. Different from SVM we used before, the non-sentiment method we use here is an end-to-end method that directly takes the raw inputs of words from news headlines and generate a prediction of the stock market. There is no manually labeled features like sentiment. In text-mining field. A lot of methods such as SVM or classification tree or logistic regression rely on the manually labeled features which heavily depends on human domain knowledge. In the aforementioned method of using sentiment to predict the stock market via SVM, we need to compose a dictionary to store all words and their corresponding labels of being either positive or negative. What if there is no pre-existing domain knowledge for this task? Is it possible to compose a method that does not need human knowledge but directly establish the link between the inputs information we have and the desired outputs we need?

In NLP (Natural Language Processing) problem, RNN (recurrent neural network) is a widely accepted deep learning method. The idea behind the RNN is to make full use of the information hidden in sequences. In a traditional text-mining methods, we usually assume that all inputs in a document or sentence are equally contributing to the task. For example, we often use features like TF-IDF of certain words to compose a vector to represent a document, despite those words appear in the document differently. The assumption that those words are independently existing among one another can be harmful. It make the algorithm fail to identify information at fine granularity, and also make it hard to

achieve high performance of certain task. The notion of recurrent neural network is not novel in artificial intelligence. It has existed for decades. Next, we would introduce a simple example of RNN and describe how we model our task by using it.

A simple RNN

A simple RNN could be described as shown in Figure 3.3-1. The left-hand part of the figure shows the recurrent neural network before being unrolled. x is an input which is an element in a sequence being processed. U is the weights being multiplied by x. s is the output of the node represented by a circle. V is the weights multiplied by the hidden state s. o is the final output of the neural network. The structure described above is a basic and classical structure of neural network. The special part about RNN the inducement of the recurrent structure. The hidden state of s in last step (t-1) is also taken as input of the neural network at time t. Thus, the network has a "memory" mechanism which further make it able to process the sequential information.

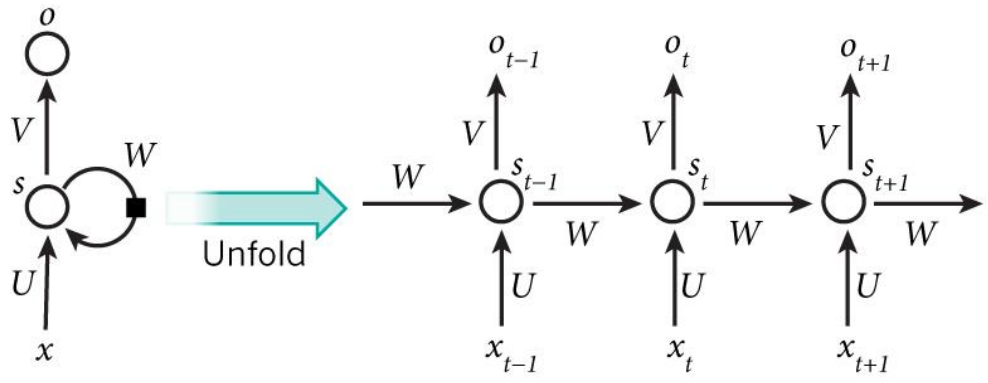

Figure 3.3-1 A simple RNN

As shown in the right-hand of the figure, the RNN can be unfolded to be a classical neural network. Actually, it is exactly how RNN can be trained. x_{t-1}, x_t, x_{t+1} are the different parts of a sequence, the subscript shows at which time step those parts get into the RNN. Regarding the different outputs of the neural network, o_{t-1}, o_t, o_{t+1} are different outputs of the network at different time step.

While only later outputs of the network contains more up-to-date information. In this case, the last output contains all information in the focal sequence being processed by the neural network.

RNN in market prediction

In our project, we utilize the RNN to predict the stock market trend. The task of using news headlines to predict the stock market trend can be seen as a classical sentence classification problem. In this section, we describe how we model the stock market prediction problem into a sentence classification problem by using RNN.

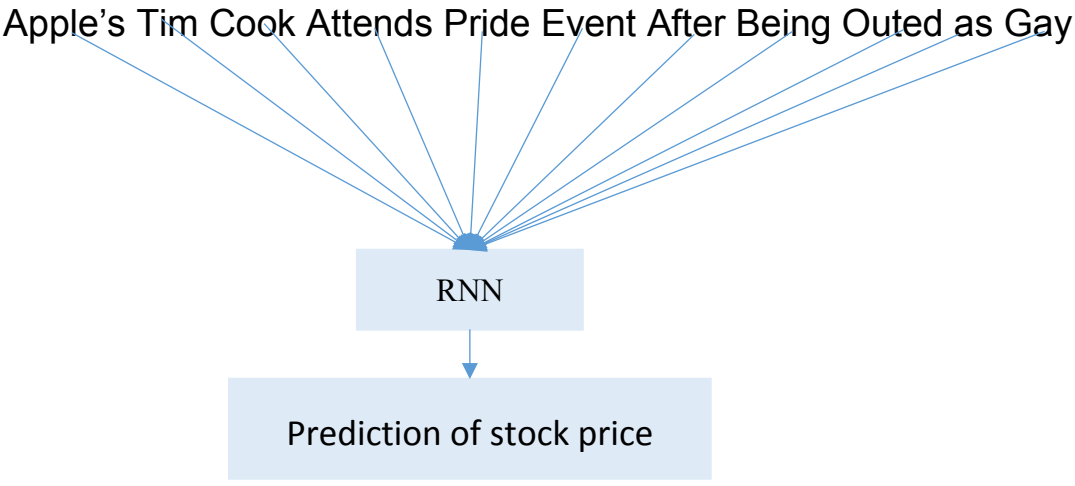

Figure 3.3-2 The RNN used for predicting stock trend

Figure 3.3-2 shows how the sentence can be put into the RNN. We treat a sentence as a sequence where each word is an element of the sequence and will be put into the RNN one by one at each time step. The output of the RNN is a real value ranging from 0 to 1. In our further evaluation of the algorithm, we arbitrarily select 0.5 as the threshold to differentiate the prediction of upward and down trend (i.e. values larger than 0.5 are deemed as a prediction of upward trend and those below 0.5 are deemed as downward trend). We further draw a Figure 3.3-3 to show the details.

Figure 3.3-3 shows how we utilize the RNN to model the prediction problem. The letters represents similar concepts as Figure 3.3-1 does. We represent each word in the sentences as one vector (bag of words). Each vector will be put into the RNN one by one till the last one. After the RNN receives the last word, it will generate a value indicating the predicted trend of the stock price. The design of RNN is described in Figure 3.3-4

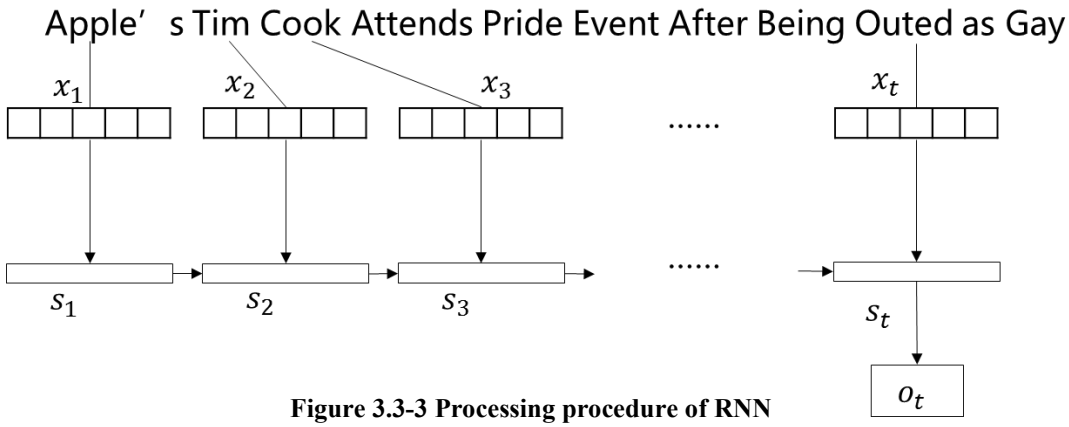

Figure 3.3-3 Processing procedure of RNN

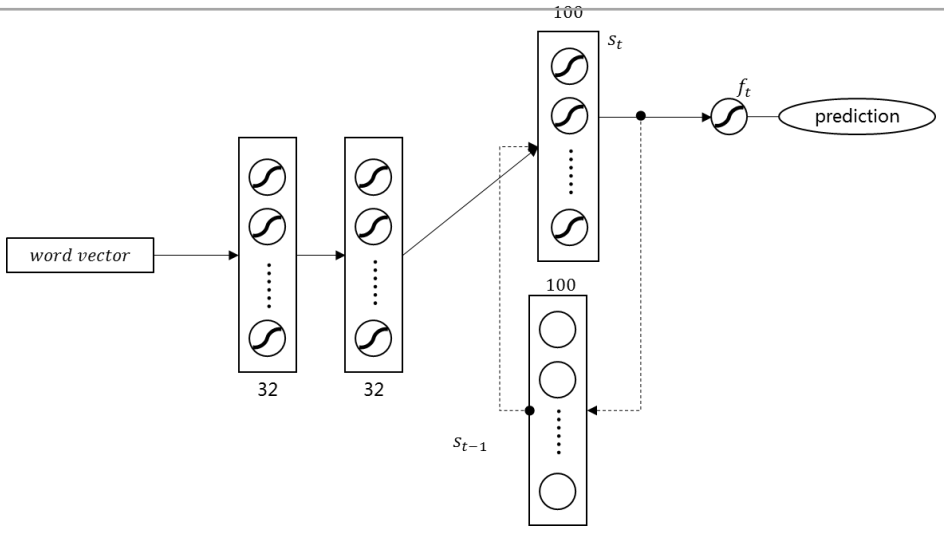

Figure 3.3-4 Structure design of RNN

The word vectors are taken as inputs. At each time, the RNN takes one vector into the first two dense layer with 32 nodes each. The third layer is a recurrent layer sized 100 nodes. The third layer not only takes the current signals generated by the first two layers at time t but receives that of t-1.

We use Adam optimizer to conduct BPTT (back propagation through time) algorithm to optimize

the neural network and deploy an early stopping mechanism to prevent overfitting problem of RNN. We use all data before 01/07/2014 as training data and use randomly selected 10% of it for early stopping mechanism. If there is no improvement achieved by the optimizer in five steps of optimization. We stop the training process of RNN and generate prediction on the test set (01/07/2014-26/08/2014).

3.4 Non-Sentiment-Based: LSTM

The LSTM (Long Short Term Memory) neural network is an advanced version of RNN. It is advanced in the sense that it not only consider the sequential information in the sequence but also can detect and learn which elements in the sequence are more important that other elements. A RNN could only take the closer elements in the sequence as more important, which make it hard to understand the complex information hidden in the sentences. The LSTM deal with this problem by introducing a memory filtering mechanism. This mechanism takes the place of the recurrent layer. The other parts of LSTM we use are identical to the RNN we introduced above. Thus, we do not describe the details of the LSTM.

4 Result

4.1 Price-Based: ARIMA

The accuracy of prediction was calculated by comparing the trend of true trend and the predicted trend. If both trends are upward or downward, this prediction was correct. The total prediction accuracy was the percentage of correct predictions in all the transaction days in July.

Table 4.1-1 (Appendix) shows the best ARIMA model for each company and the corresponding prediction accuracy, AIC and BIC. The evaluation of each ARIMA model was based on the criteria of smallest AIC or BIC. The results of each 10 different ARIMA model are attached as a table in the appendix.

4.2 Sentiment-Based: SVM

In sentiment-based stock market prediction, we mention two methods. The first sample method assume that positive news will lead to upward trend while negative news will lead to downward trend. The second model use SVM to automatically find out the correlation patterns between sentiments and prices trend.

Table 4.2-1(Appendix) shows the results of the first prediction method. It shows that most of results are near to 50% and worse than that of ARIMA models.

Table 4.2-2 (Appendix) shows the results of sentiment-based SVM model. We predict the price trend of the day, the first day, the second day and the third day after the publication of the news. The average accuracy represents the average values of the 0, 1, 2 and 3 lag. We will attach our detail results in the attachment. From Table 4.2-2, we know that the accuracy of most companies surpasses the first method as well as the ARIMA model.

4.3 Non-Sentiment-Based: RNN

In this section, we report the experiment results of the RNN on predicting the 7 firms' stock price changes using three different types of news (technical news/all relevant technical news/all relevant news) with 0,1,2,3 days lag.

In Table4.3-1 (Appendix), we show some example prediction using RNN.

The predicted values are direct outputs of the RNN. We arbitrarily select 0.5 as threshold to differentiate the prediction of upward trend or down ward trend. For example, the value of 0.15 predicted by RNN by reading the first piece of news headline indicates a prediction of downward trend of today's price of Apple. The real value of today's price change is 0 which indicates a downward trend of today's price.

Table 4.3-2 (Appendix) summarizes the results of the experiments. Each bock of three rows summarizes results of the results on one firm (one stock).different inputs means different inputs of news types, the corresponding results with 4 different lags are shown in the right block. The average accuracy is calculated based on the accuracy of all lags using one specific news type. The bold ✔ in the left-hand block shows which news type achieved the best performance on all different lags. The bold red numbers in the right-hand block shows prediction with which lag achieve the best performance.

Based on Table4.3-1, we analyze the results and find some interesting findings.

Overall speaking, we achieved better performance than ARIMA which only use price information as inputs. This finding serves as evidence that news information indeed could be used as reference for prediction even without knowing anything about the prices.

In terms of different news types, results show that only the type of all relevant news and all technical news show good performance. The input of all relevant technical news cannot generate competitive prediction compared with two other inputs. This finding is contradictory against what has been found in the experiments of SVM. The reason of it may be the lack of data. RNN or other kinds of neural networks typically requires a lot of data to generate excellent performance.

In terms of different time lags of prediction, we find that the news headline predict the stock market with a commonly existing 2-3 days lag. For most experiments, the best performance is achieved with 2-3days lag. Except form Google and Microsoft, almost all prediction of other firms' stock prices trends show best performance on the third or the fourth day.

4.4 Non-Sentiment-Based: LSTM

We report on the results of experiments using LSTM in a similar way. Table 4.4-1 (Appendix)

summarizes the results using LSTM.

We get similar findings compared with those from experiments using RNN. First, the technical relevant news still cannot generate equivalent performance to other two news types. And almost all prediction shows the best performance with a 2-3 days lag. The Microsoft stock shows the best performance when being predicted with 2-3 days lag, which is different from the results of RNN. Only the stock of Google can be predicted with a shorter lag compared with other stocks, which is consistent with the results of using RNN.

5 Discussion and Conclusion

In this study, we investigate how to extract useful information from news headline and utilize it to predict financial market. As the benchmark, we firstly utilize time series method, ARIMA model, to predict price trend of seven technology firms, which are Apple, Amazon, Facebook, Google, Microsoft, IBM, and Tesla, using only stock price data. In the contrary, we also employ three machine learning methods using only news headlines to conduct the price trend prediction task for the same seven firms. The three machine learning methods include Support Vector Machine (SVM), Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM) with one basic difference, which is that we use sentiment analysis to support SVM but not for the last two neural network models.

In our study, we can define the seven technology firms as F_j $(j = 1, \dots, 7)$, the ARIMA model using only price for the firm F_j as M_j^P, and the models using news headline for the firm F_j as $M_{i,j}^H$ with $i = 1,2,3$. For the model selection of ARIMA model using price, M_j^P, we use Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to select the preferred parameters and preferred model M_j^{P*} among a set of candidate models. As for the model using headline, $M_{i,j}^H$, we use two parameters to characterize them: 1) Different news types; 2) Different days of time lag. Then we

can further define the model using news headline for the firm F_j as $M_{i,j}^H(NT_k, L_l)$, where NT_k $(k = 1,2,3)$ represents the three news types and L_l $(l = 1,..,4)$ represents the time lags from 0 days to 3 days. To select the preferred parameters and preferred model among a set of candidate models, we can select the model $M_{i,j}^H$ with the highest accuracy in regard to the news type NT_k and time lag L_l, which can be expressed as

$$Accuracy(M_{i,j}^H(NT^*, L^*)) = max_{NT_k, L_l} \text{Accuracy}(M_{i,j}^H(NT_k, L_l)),$$

where $M_{i,j}^H(NT^*, L^*)$ is the preferred model and NT^*, L^* are the corresponding preferred parameters. To select the best model for the firm F_j, we can select the model with highest accuracy among the three machine learning models, which can be expressed as

$$\text{Accuracy}(M_j^{H*}(NT^*, L^*)) = max_i \text{Accuracy}(M_{i,j}^H(NT^*, L^*)),$$

where $M_j^{H*}(NT^*, L^*)$ is the best model for the firm F_j and NT^*, L^* are the corresponding preferred parameters.

In Table 5-1(Appendix) you can find the combined result of our experiment, where accuracy in bold represents the accuracy of the preferred model $M_{i,j}^H(NT^*, L^*)$ and accuracy in red represents the accuracy of best model $M_j^{H*}(NT^*, L^*)$. Through the comparison of the results of different methods in our experiment, we can draw the four conclusions in the following parts.

5.1 News headline can be used to predict financial market

As shown in Table 5.1-1(Appendix), through the comparison between the accuracy of preferred ARIMA model using only price for the frim F_j, M_j^{P*}, and the accuracy of the best model using only news headline for the same firm F_j, $M_j^{H*}(NT^*, L^*)$, we can find that methods using only news headline outperform the method using only price significantly for every firm. In Table 5-1, we can also find that even for the inferior models using news headline, almost all of them outperform ARIMA

model for every firm. And the best model using only news headline for the firm F_j, $M_j^{H*}(NT^*, L^*)$, is really precise (around 80%) with average accuracy of 84%.

Thus we can conclude that news headline can be used to predict financial market with considerable performance. In the future study, we can consider using the combination of price and textual data, such as news headline, to build more accurate model to predict financial market trend.

5.2 About different models using news headline

In our study, we use sentiment extracted from news headline to support SVM and as for RNN and LSTM, we don't use sentiment analysis. Comparing the preferred models using headline for the firm F_j, $M_{i,j}^{H}(NT^*, L^*)$ with $i = 1,2,3$ (the accuracy in bold in Table 5-1), we can find that for every firm, the best model $M_j^{H*}(NT^*, L^*)$ is always SVM model supported by sentiment. No matter what news type is used, SVM can obviously performs better than RNN and LSTM. However, lots of studies have shown that RNN and LSTM models are supposed to show better performance than SVM in different tasks such as facial recognition, speech translation, and financial prediction. As shown in Table 3.2-1, sentiment and stock price trend shown considerable correlation which means that sentiment is one factor that can be used to predict price trend. We believe that one of the main reasons why SVM outperforms RNN and LSTM in our study is that sentiment extracted from news headline give a great help in the trend prediction task. Another reason might be the number of layers of RNN and LSTM in our experiment is not large enough. We use three layers in RNN and LSTM and neural network models are believed to show better performance when they are "deep" enough.

In a word, we can say that sentiment-based methods, such as SVM in our study, outperforms non-sentiment-based methods, such as RNN and LSTM in our study. In the future work, we can consider embedding sentiment analysis into RNN/LSTM model and increase the number of layers in

RNN/LASTM model to improve the performance.

5.3 About different news types

In our study, we find that different news types showed different predictive power. We use three different news types: firm relevant news, all technical news, and firm relevant technical news. In Table5.1-1, we can find that for the best model, $M_j^{H*}(NT^*, L^*)$ of the firm F_j, the corresponding news type NT^* is quite different. For the firms such as Apple, Amazon, Facebook, Google, the news type of their best model is All Relevant news. And for the firms such as Apple, Microsoft, IBM and Tesla, the news type of their best model is Tech Relevant news. There is no firm that the news type of its best model is Tech News, which means that only news headline that is relevant to the firm can makes significant difference in financial market trend prediction.

However, there still exist some limitations about the news type in our study. We use the filter of key word to extract the relevant news. But frankly speaking, this kind of method is not really precise enough because there exist so many ambiguous case, such as Apple could be the name of the firm or the name of the fruit. And moreover, some headlines might not mention the name or the ticker of the firm and we can only ignore such kind of case. In the future work, we can develop more precise filter of relevant news using some techniques of NLP.

5.4 About different days of lag

We find that there exist time lags in financial market trend prediction when using news headline. Table 5.4-1(Appendix) show the corresponding days of time lag L^* for the preferred model of the firm F_j, $M_{i,j}^H(NT^*, L^*)$ and the days of time lag for the correlation between the sentiment value and price trend. We can find that for same news type NT^* and firm, the days of time lag are consistent. After calculating the average of days of time lag for each firm, we can find that different firms show

different days of time lags. And we can rank the firms in according to the days of time lag from the largest to the smallest: Apple, Amazon > Microsoft, IBM > Facebook > Tesla > Google. Google is the firm with smallest days of time lag, which is only 0.83 days. We can regard Google as the firm with quickest response to the market information.

The possible reasons to the difference could be: 1) Different investors groups for different firms. For example, younger investor group might show faster respond speed to the news than the elder investor group. Google might have younger investor group than other firms. 2) Special news was reported during observational period for some firms, e.g., earning announcement, new product release. Special news might draw more attentions from the investors and could lead to more immediate investment reaction. We can conduct more studies on these in detail in the future.

5.5 Summary

In this study, we conduct ARIMA model using only price and three machine learning method using only news headline (SVM, RNN, and LSTM) to predict the price trend of seven different technology firms. Through the experiment and analysis of result, we find four main conclusions about the practical usefulness of news headline in financial market prediction, the relation of sentiment analysis and financial market prediction, importance of the relevance of news type, and the response speed to market of different firms. Due to the time limit, there still exist some limitations in our study and there exist some possible approaches we will continue to pursue.

Reference

Baker, M. and Wurgler, J. (2006) 'Investor Sentiment and the Cross‐Section of Stock Returns', *The Journal of Finance*, *61*(4), 1645-1680.

Nofsinger, J. R. (2005) 'Social Mood and Financial Economics', *Journal of Behavioral Finance 6*(3), 144-160.

Strapparava, C. and Mihalcea, R. (2008) 'Learning to identify emotions in text',(Ed.)^(Eds.), *SAC '08 Proceedings of the 2008 ACM symposium on Applied computing*, Fortaleza, Ceara, Brazi.

TETLOCK, P. C. (2007) 'Giving Content to Investor Sentiment: The Role of Media in the Stock Market', *The Journal of Finance*,

62(3), 1139-1168.

Appendix

Table 3.2-1. Correlation between sentiment and stock price trend

Company	News type	Same day	One day lag	Two day lag	Three day lag
Apple	All relevant news	0.061	-0.093	**-0.278**	-0.191
	All tech news	-0.084	-0.010	**-0.142**	0.041
	All relevant tech news	-0.009	-0.123	**-0.247**	-0.180
Amazon	All relevant news	-0.002	-0.132	-0.078	**0.230**
	All tech news	-0.135	0.096	**-0.233**	0.062
	All relevant tech news	-0.106	-0.090	-0.023	**0.155**
Facebook	All relevant news	0.031	-0.001	0.106	**0.116**
	All tech news	0.022	**0.084**	-0.072	-0.060
	All relevant tech news	0.090	**-0.190**	0.152	0.066
Google	All relevant news	**-0.209**	0.068	0.144	-0.012
	All tech news	**-0.104**	0.009	-0.857	0.035
	All relevant tech news	**-0.303**	0.029	0.068	0.078
IBM	All relevant news	-0.008	-0.046	0.057	**0.301**
	All tech news	-0.137	-0.013	0.026	**0.207**
	All relevant tech news	-0.005	0.061	-0.091	**0.340**
Microsoft	All relevant news	0.028	**0.421**	-0.190	-0.073
	All tech news	-0.039	**0.185**	0.081	-0.025
	All relevant tech news	0.011	0.018	**-0.069**	0.013
Tesla	All relevant news	**-0.172**	0.041	0.161	0.073
	All tech news	0.083	**-0.094**	-0.034	0.014
	All relevant tech news	-0.040	0.204	**0.402**	0.223

Table 4.1-1 Prediction accuracy of best ARIMA models

Company	Predication Accuracy	Model	AIC	BIC
Apple	45.00%	ARIMA(1, 1, 0)	**222.4748**	**227.1883**
Amazon	45.00%	ARIMA(1, 1, 1)	**520.9218**	**527.9919**
Facebook	22.50%	ARIMA(2, 1, 2)	**288.9593**	300.7428
	47.50%	ARIMA(0, 1, 1)	291.546	**296.2594**
Google	65.00%	ARIMA(1, 2, 1)	**555.2298**	**562.2612**
	65.00%	ARIMA(2, 2, 1)	**555.2298**	**562.2612**
Microsoft	40.00%	ARIMA(2, 1, 2)	**86.44131**	98.22485
	55.00%	ARIMA(1, 1, 1)	87.40701	**94.47714**
IBM	50.00%	ARIMA(2, 1, 1)	**303.6116**	313.0384
	42.22%	ARIMA(0, 1, 1)	304.986	**309.6994**
Tesla	42.22%	ARIMA(0, 1, 1)	**512.3571**	**517.0705**

Table 4.2-1 Prediction accuracy of sentiment prediction model without machine learning methods

Company	Hit rate based on relevant news	Hit rate based on tech relevant news	Hit rate based on tech news
Apple	54.17%	50.72%	52.86%
Amazon	50.77%	48.00%	45.71%
Facebook	50.70%	51.72%	58.57%
Google	45.21%	46.38%	51.43%
IBM	50.00%	53.33%	38.03%
Microsoft	47.76%	49.21%	44.29%
Tesla	38.71%	47.62%	60.00%

Table 4.3-1. Example outputs of the RNN

Date	Headlines	0 day lag		1 day lag		2 days lag		3 days lag	
		predicted	real	predicted	real	predicted	real	predicted	real
7/1/2014	Apple CEO Tim Cook Attends Gay Pride Parade After Being 'Outed' By CNBC ...	0.15	0	0.04	0	0.83	1	0.29	0
7/1/2014	San Francisco gay pride parade attracts thousands and Apple's Tim Cook	0.24	0	0.28	0	0.42	1	0.29	0
7/1/2014	AAPL: Apple CEO Tim Cook Marches in Gay Pride Parade	0.52	0	0.07	0	0.43	1	0.41	0
7/1/2014	Apple's Tim Cook Attends Pride Event After Being Outed as Gay	0.18	0	0.17	0	0.30	1	0.21	0
7/1/2014	Tim Cook shows 'Apple Pride' at gay pride parade	0.10	0	0.16	0	0.69	1	0.42	0
7/1/2014	Apple, Facebook, Google cheer on Gay Pride parade	0.21	0	0.43	0	0.58	1	0.48	0

Table 4.2-2 Prediction accuracy of sentiment-based SVM model

Stock	Data Type	Average Accuracy
Apple	AAPL Apple news	72.6%
	AAPL Apple tech news	72.6%
	AAPL tech news	**73.1%**
Amazon	AMZN Amazon news	59.6%
	AMZN Amazon tech news	**63.6%**
	AMZN tech news	61.4%
Facebook	FB Facebook news	**79.5%**
	FB Facebook tech news	78.5%
	FB tech news	69.6%
Google	GOOG google news	70.6%
	GOOG google tech news	70.2%
	GOOG tech news	**73.1%**
IBM	IBM IBM news	72.3%
	IBM IBM tech news	**75.4%**
	IBM tech news	60.7%
Microsoft	MSFT microsoft news	68.6%
	MSFT microsoft tech news	68.7%
	MSFT tech news	**70.8%**
Tesla	TSLA Tesla news	**66.6%**
	TSLA Tesla tech news	61.4%
	TSLA tech news	66.3%

Table 4.3-2 Prediction accuracy on different stocks (RNN)

Stock	Inputs			Accuracy				
	All Relevant News	Tech Relevant News	Tech News	Average Accuracy	0 day lag	1 day lag	2 days lag	3 days lag
Apple	✔			62.0%	55.3%	58.6%	59.5%	**74.7%**
		√		59.2%	56.5%	54.8%	57.4%	**68.1%**
			√	59.8%	59.8%	55.1%	58.4%	**65.8%**
Amazon	√			54.4%	51.0%	50.7%	43.9%	**72.0%**
		√		52.6%	53.2%	44.9%	43.3%	**68.9%**
			✔	56.6%	50.2%	56.8%	**59.9%**	59.7%
Facebook	✔			70.2%	57.9%	62.8%	**88.5%**	71.8%
		√		66.6%	55.8%	59.6%	**80.6%**	70.5%
			√	64.1%	56.3%	63.6%	**71.1%**	65.2%
Google	√			61.0%	56.6%	**66.1%**	63.4%	58.0%
		√		54.7%	**58.7%**	55.5%	50.9%	53.6%
			✔	64.8%	59.3%	64.1%	**68.0%**	67.9%
IBM	√			47.1%	25.0%	56.9%	**81.8%**	24.6%
		√		51.8%	21.6%	53.0%	**78.5%**	53.9%
			✔	53.1%	43.4%	50.9%	58.4%	**59.7%**
Microsoft	✔			63.8%	54.1%	**71.9%**	66.1%	63.3%
		√		63.1%	51.8%	**69.6%**	66.9%	64.0%
			√	58.8%	52.6%	55.9%	61.3%	**65.6%**
Tesla	✔			60.7%	43.1%	52.7%	**82.9%**	64.1%
		√		52.0%	42.7%	52.4%	**56.7%**	56.1%
			√	57.7%	56.7%	**65.9%**	62.5%	45.7%

Notes: the table summarized the results of all sets of experiments. Each bock of three rows summarizes results of the results on one firm (one stock).different inputs means different inputs of news types, the corresponding results with 4 different lags are shown in the right block. The average accuracy is calculated based on the accuracy of all lags using one specific news type. The bold ✔ in the left-hand block shows which news type achieved the best performance on all different lags. The bold red numbers in the right-hand block shows prediction with which lag achieve the best performance.

Table 4.4-1 Prediction accuracy on different stocks (LSTM)

Stock	Inputs			Accuracy				
	All Relevant News	Tech Relevant News	Tech News	Average Accuracy	0 day lag	1 day lag	2 days lag	3 days lag
Apple	✔			**61.8%**	56.6%	56.7%	59.5%	**74.33%**
		√		60.8%	61.1%	53.4%	54.2%	**74.34%**
			√	59.3%	54.9%	57.2%	58.5%	**66.56%**
Amazon	√			54.4%	54.0%	51.1%	42.0%	**70.60%**
		√		49.6%	57.4%	41.3%	41.7%	**57.95%**
			✔	**56.3%**	49.3%	56.2%	59.3%	**60.43%**
Facebook	✔			**68.9%**	58.0%	55.8%	**88.1%**	73.79%
		√		68.5%	57.2%	60.7%	**83.3%**	72.76%
			√	63.3%	56.4%	63.5%	**70.6%**	62.64%
Google	√			57.7%	58.6%	**64.2%**	54.3%	53.53%
		√		56.7%	**62.1%**	58.1%	49.5%	57.12%
			✔	**63.3%**	59.7%	62.6%	**67.6%**	63.31%
IBM	√			47.1%	25.0%	56.9%	**81.8%**	24.62%
		√		52.7%	15.2%	53.0%	**78.5%**	64.19%
			✔	**53.0%**	44.7%	51.0%	57.6%	**58.70%**
Microsoft	✔			**63.5%**	54.6%	67.5%	**68.3%**	63.33%
		√		62.0%	52.0%	65.2%	**67.9%**	62.84%
			√	59.5%	54.0%	56.2%	61.7%	**66.25%**
Tesla	√			57.3%	42.1%	40.0%	**82.9%**	64.18%
		√		57.6%	42.5%	52.4%	**82.5%**	53.14%
			✔	**60.7%**	58.2%	**65.4%**	61.9%	57.29%

Notes: the table summarized the results of all sets of experiments. Each bock of three rows summarizes results of the results on one firm (one stock).different inputs means different inputs of news types, the corresponding results with 4 different lags are shown in the right block. The average accuracy is calculated based on the accuracy of all lags using one specific news type. The bold ✔ in the left-hand block shows which news type achieved the best performance on all different lags. The bold red numbers in the right-hand block shows prediction with which lag achieve the best performance.

Table 5-1 Prediction Accuracy for different model of different Firms, different News Types and Time of Lags

Firms		ARIMA(p,d,q)	All Relevant				Tech Relevant				Tech News			
			Cor(Sent, Trend)	SVM	RNN	LSTM	Cor(Sent, Trend)	SVM	RNN	LSTM	Cor(Sent, Trend)	SVM	RNN	LSTM
Apple	**Lags**	(1,1,0)	2	**3**	**3**	3	2	**3**	3	**3**	2	3	3	3
	Accuracy	45%	-	**90%**	**75%**	74%	-	**90%**	68%	**74%**	-	84%	66%	67%
Amazon	**Lags**	(1,1,1)	3	3	**3**	**3**	3	**2**	3	3	2	3	2	3
	Accuracy	45%	-	73%	**72%**	**71%**	-	**78%**	69%	58%	-	68%	60%	60%
Facebook	**Lags**	(0,1,1)	3	**1**	**2**	**2**	1	1	2	2	1	2	2	2
	Accuracy	48%	-	**92%**	**89%**	**88%**	-	91%	81%	83%	-	74%	71%	71%
Google	**Lags**	(1,2,1) or (2,2,1)	0	**1**	1	1	0	1	0	0	0	2	**2**	**2**
	Accuracy	65%	-	**81%**	66%	64%	-	78%	59%	62%	-	75%	**68%**	**68%**
Microsoft	**Lags**	(1,1,1)	1	1	2	**2**	2	**1**	**1**	2	1	3	3	3
	Accuracy	55%	-	82%	66%	**68%**	-	**82%**	**70%**	68%	-	77%	66%	66%
IBM	**Lags**	(2,1,1) or (0,1,1)	3	2	**2**	**2**	3	**0**	2	2	3	3	3	3
	Accuracy	50%	-	82%	**82%**	**82%**	-	**85%**	79%	79%	-	70%	60%	59%
Tesla	**Lags**	(0,1,1)	0	2	**2**	**2**	2	**2**	2	2	1	1	1	1
	Accuracy	42%	-	83%	**83%**	**83%**	-	**83%**	57%	83%	-	76%	66%	65%
Average Accuracy		-	-	83%	**76%**	**76%**	-	**84%**	69%	72%	-	75%	65%	65%

Notes: 1. There are three types of news: a) All Relevant - All news relevant to the firm; b) Tech Relevant - All the news relevant to the firms and the news category is technology; c) Tech News - All the news with news category in technology;
2. Accuracy numbers in bold represents that it is the best performance of according model;

Table5.1-1 Comparison between method of using price (ARIMA) and best model among the methods of using headlines

Firms		News Type	Model	Lags	Accuracy
Apple	Using price	-	ARIMA(1,1,0)	-	45%
	Using headlines	All Relevant or Tech Relevant	SVM	3	**90%**
Amazon	Using price	-	ARIMA(1,1,1)	-	45%
	Using headlines	All Relevant	SVM	2	**78%**
Facebook	Using price	-	ARIMA(0,1,1)	-	50%
	Using headlines	All Relevant	SVM	1	**92%**
Google	Using price	-	ARIMA(1,2,1) or ARIMA(2,2,1)	-	64%
	Using headlines	All Relevant	SVM	1	**81%**
Microsoft	Using price	-	ARIMA(1,1,1)	-	55%
	Using headlines	Tech Relevant	SVM	1	**82%**
IBM	Using price	-	ARIMA(2,1,1) or ARIMA(0,1,1)	-	50%
	Using headlines	Tech Relevant	SVM	0	**85%**
Tesla	Using price	-	ARIMA(0,1,1)	-	45%
	Using headlines	Tech Relevant	SVM	2	**83%**
Average Accuracy	Using price				51%
	Using headlines				**84%**

Table 5.4-1 Comparison of time lag between different models

Firms		All Relevant					Tech Relevant					Tech News					Ave. Lag for 3 News Type
		Cor(S,T)	SVM	RNN	LSTM	Ave. Lag	Cor(S,T)	SVM	RNN	LSTM	Ave. Lag	Cor(S,T)	SVM	RNN	LSTM	Ave. Lag	
Apple	Lags	2	3	3	3	2.75	2	3	3	3	2.75	2	3	3	3	2.75	2.75
	Accuracy	-	90%	75%	74%	-	-	90%	68%	74%	-	-	84%	66%	67%	-	-
Amazon	Lags	3	3	3	3	3.00	3	2	3	3	2.75	2	3	2	3	2.50	2.75
	Accuracy	-	73%	72%	71%	-	-	78%	69%	58%	-	-	68%	60%	60%	-	-
Facebook	Lags	3	1	2	2	2.00	1	1	2	2	1.50	1	2	2	2	1.75	1.75
	Accuracy	-	92%	89%	88%	-	-	91%	81%	83%	-	-	74%	71%	71%	-	-
Google	Lags	0	1	1	1	0.75	0	1	0	0	0.25	0	2	2	2	1.50	0.83
	Accuracy	-	81%	66%	64%	-	-	78%	59%	62%	-	-	75%	68%	68%	-	-
Microsoft	Lags	3	1	2	2	2.00	3	1	1	2	1.75	1	3	3	3	2.50	2.08
	Accuracy	-	82%	66%	68%	-	-	82%	70%	68%	-	-	77%	66%	66%	-	-
IBM	Lags	1	2	2	2	1.75	2	0	2	2	1.50	3	3	3	3	3.00	2.08
	Accuracy	-	82%	82%	82%	-	-	85%	79%	79%	-	-	70%	60%	59%	-	-
Tesla	Lags	0	2	2	2	1.50	2	2	2	2	2.00	1	1	1	1	1.00	1.50
	Accuracy	-	83%	83%	83%	-	-	83%	57%	83%	-	-	76%	66%	65%	-	-
Average Lag		1.71	1.35	1.45	1.45	1.96	1.86	1.13	1.27	1.36	1.79	1.43	1.59	1.47	1.54	2.14	-