Stock Market Prediction Exploiting Microblog Sentiment Analysis

Bo Zhao, Yongji He, Chunfeng Yuan, and Yihua Huang
National Key Laboratory for Novel Software Technology, Nanjing University, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China
{chawbhoppi, riskyhe}@smail.nju.edu.cn, {cfyuan, yhuang}@nju.edu.cn

***Abstract*—This paper proposes a stock market prediction method exploiting sentiment analysis using financial microblogs (Sina Weibo). We analyze the microblog texts to find the financial sentiments, then combine the sentiments and the historical data of the Shanghai Composite Index (SH000001) to predict the stock market movements. Our framework includes three modules: Microblog Filter (MF), Sentiment Analysis (SA), and Stock Prediction (SP). The MF module is based on LDA to get the financial microblogs. The SA module first sets up a financial lexicon, then gets the sentiments of the microblogs obtained from the MF module. The SP module proposes a user-group model which adjusts the importance of different people, and combines it with stock historical data to predict the movement of the SH000001. We use about 6.1 million microblogs to test our method, and the result demonstrates our method is effective.**

I. INTRODUCTION

Stock prediction has become popular with the rise of internet finance. The EMH (Efficient Market Hypothesis) suggests that stock market fully reflects all available information, and follows a random pattern. This hypothesis was widely accepted by economists over the past years, but now people believe that we can depict a tip of the stock market. Many methods from other fields have shown promising results in this area. With the advantages of the sufficient online data and the AI methods, some researchers apply NLP (natural language processing) techniques in news [1] [2], message boards [3] [4] and tweets [5] [6] to train a certain pattern to cover the stock market movement. These works use many different strategies and achieve remarkable results. But there still need some efforts to improve the performance. Our paper extends the research by adding the microblog filter and user-group model, which are proven effective to predict the stock market movement.

Using social sentiments to predict stock market movement is initiated by Bollen and Mao [7]. Inspired by their work, sentiment begins to play an important role in stock prediction research. The investor's sentiment theory also states that the trader's investment behavior will be influenced by sentiment [8]. Previous studies focus on proposing new algorithms to extract sentiment with microblogs [6] [9] [10]. Si et al [6] use the topic sentiment to analyze the stock market; Marta et al [9] and Chong et al [10] use lexicon method to identify the sentiment of microblog texts, and aggregate the sentiments as a daily index to predict the stock price.

Many related researches pay more attention to the microblog text analysis and the prediction method for stock market, while the data filtering does not draw much attention. As the raw data is copious and noisy, loading all of the data will be a waste of time and even disturb the outcome. Therefore, we need to clean the raw data. There are several ways to clean microblogs: symbol-tagged filter [10], accounts filter [9] and keywords filter. The mostly used method is keywords filter. But keywords filter is not competent in this task, because of polysemant, proper names, idioms, and especially internet slangs. We resolve this problem by using LDA (Latent Dirichlet Allocation) model and keyword list to generate a representative vector of the microblog, and the multi-field comparison is applied to discover whether a microblog is related to the finance or not. Marta et al [9] also use LDA to clean the microblogs. But the target of the LDA is to distinguish the microblogs from some specific accounts. This is also not competent for the task, so we propose a multi-field comparison method based on LDA to clean microblog data.

It is a common method that aggregating sentiments of all users to predict stock market. But we all know that sentiments from different users have different influences. Some users are experts or have foresight in stock market, and mostly their posters are sensitive and delicate to stock market, while some are newcomers and obtuse to stock market movement. To distinguish these two kinds of users will help us in developing an effective model to predict stock market. For this purpose, we conduct a user classification process and propose a user-group model. Roy et al [5] also classify users into two categories. Although we all divide the users into two categories, the category definitions, the classification methods, especially the aims are different. The aim of [5] is to find the experts and discard other users. We believe that the users who are not experts also have influences on stock market. So we propose a new user-group model, which takes the information from all users. Besides, by involving all the information and getting it weighted, the model also relieves the deviation of previous processing stages.

In this paper, we consider sentiment analysis as the significant junction of text data and stock market. We design a method to predict stock market movement, and propose an effective method based on LDA model to clean microblogs, and a user-group model to predict stock market movement. Our contributions are: (1) develop a microblog filtering module, which can efficiently distinguish the financial microblogs from others; (2) create a financial sentiment lexicon, which turns out

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

to be useful for financial microblogs sentiment analysis, (3) develop a combined method for sentiment analysis based on the lexicon, feature selection and word embedding, which achieves a progressive performance; (4) propose a user-group model which subtly displays the user profiles and reduces the deviation of the previous processing stages of our method, and combine the model with stock historical data for stock prediction.

The rest of the paper is organized as follows. Section II describes our dataset that we use for this paper and details our methods, including the Microblog Filter (MF) module, the Sentiment Analysis (SA) module and the Stock Prediction (SP) module. In section III, we conduct some experiments to test our methods. Finally, section IV concludes this and future work.

II. DATA AND METHOD

This section details the data and methods. This work relates to many techniques and includes three modules, the MF module, the SA module and the SP module. The MF module deals with the raw data to get the related financial microblogs. The SA module is used to extract sentiments from microblogs. Then the SP module is applied to predict the stock market movement with the extracted sentiments and the historical stock data.

A. Data Dscription

There are two types of data we need: text data and stock data. Text data is the microblog text and used to analyze the sentiment, while the stock data helps to predict the stock market. The sources and details of the two types' data are described in the following part.

The text data comes from three sources: extracted from weibo.com, downloaded from datatang (a data exchange site) and shared by pennyjob. All the data is segmented by NLPIR, an open source software developed by Huaping Zhang. The dataset from datatang contains 1.3 million microblogs and will be only used to train LDA model, since it does not contain user ID or timestamp. The data from weibo.com and pennyjob is the time series data, and contains all features we need. It is mainly used to analyze the microblog sentiments to predict stock market movement. The data is collected over the period from June to December in 2015, which provides us with 6.1 million microblogs. While the span of June to August witnesses a normal stock market alteration, so we just use this part of data to train LDA and create the financial sentiment lexicon.

The stock data comes from NetEase Money site (http://quotes.money.163.com/stock). The SH000001 index (Shanghai Composite Index) is chosen as the indicator of stock market movement. This index includes all the stocks of public companies in China and is an authoritative index published by the SSE (Shanghai Stock Exchange). This data spans a period of September 11 to December 31. For each record, just the closing price, volume, and the change amount are downloaded. We transform the price into the movement direction (up and down), i.e. if in one day its closing price is bigger than previous day, we label it as "up", otherwise, label it as "down".

As illustrated by Figure I, we start by developing a filter model to select the relevant microblogs on finance. We use LDA model to extent the semantics of the microblogs, and determine whether the microblogs are financial microblogs. In order to fully extract the sentiments of the microblogs, we use two methods: lexicon and support vector machine (SVM). Lexicon takes advantage of the finance domain, and SVM is robust to the free style of microblog text. In the SP module, we develop a user-group model, which can tolerate the noise and deviation from previous stages. We divide users into two categories. Different categories are handled by different methods.

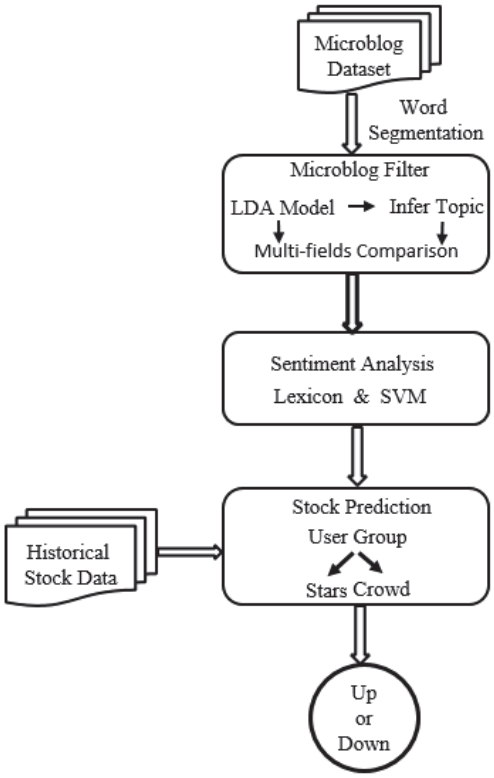

Fig. 1. The Pipeline of Our Methodology

B. Microblog Filter

For illustrating our method, we first introduce two techniques: 1) LDA is a generative probabilistic model for topic modelling [11]. It is an unsupervised algorithm mostly used to produce a word-topic distribution. And the word-topic distribution is treated as a given parameter to infer the topics of new documents. 2) Vector space model (VSM) is an algebraic model for representing text documents [12]. This method is based on bag-of-words. It needs a vocabulary list, and each term in the vocabulary represents one feature. The text vector is built by filling 1 or 0 in features, according to the fact that whether the text contains the feature term. Here, we use LDA model and VSM to clean microblogs.

We develop a meticulous method to clean the microblogs. First, train LDA model to get the word-topic distribution with aforementioned about 1.3 million microblogs from datatang and

our extraction data, totally 2.6 million microblogs. Second, select some microblogs related to several small fields as the fields' reference microblogs, each field contains 10-20 microblogs. Here, the field represents a minimum unit that has a certain topic or a unit that has a significant topic, such as stock, securities, consumption etc. Third, use the model generated from first step to infer the topics of the field microblogs and the new microblogs. Fourth, create the VSM of the microblog by the keyword list. Then merge the two vector as the final topic vector. At last, compare the final topic vector (the topic vector of new microblog) to the vectors of each field. If their similarity is within a threshold, keep the new microblog, otherwise drop it. The work flow is shown as *Figure II*.

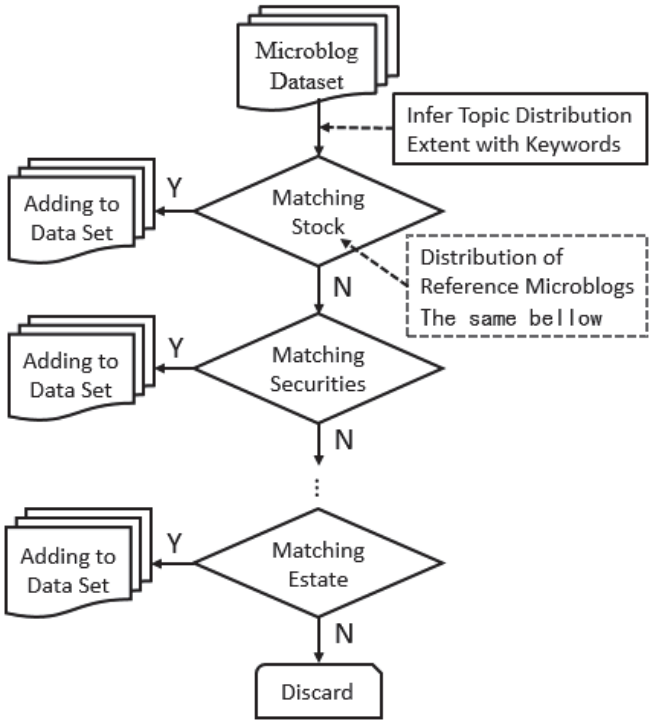

Fig. 2. Work Flow of Microblog Filter Module

This method has two assumptions: 1) the training set and the test dataset follow the same word-topic distribution, 2) these reference microblogs that belong to a certain field can cover the field topic. For the first assumption, there may have a concept drift problem after a long interval of time, but during a certain short interval, the distribution can be treated as invariant. When concept drift problem occurs, the distribution can be retrained by using the recent past data, then this problem is relieved simply. As for the assumption 2, according to our definition, there is a one-to-one correspondence between fields and topics. If these reference microblogs belong to the fields we focus on, then they can cover the corresponding topics.

C. Sentiment Analysis

In this module, we create a financial sentiment lexicon to analyze the sentiment. We also improve the performance of SVM for sentiment analysis by feature selection and word embedding.

After the process of the MF module, we get the financial microblogs and push them into the SA module. The goal of the SA module is to determine whether the sentiments of these microblogs are positive or negative. In the SA module, we use two methods to classify the sentiment. Since all the selected microblogs are in financial domain, we first consider the lexical method [10] [14]. A sentiment lexicon in financial domain is created for this method. Second, we treat the sentiment analysis as a binary classification, and use the SVM for this analysis [9]. Finally, in order to achieve a better performance, we combine the two methods.

1) Sentiment lexicon: We create a lexicon based method to extract the sentiment form microblogs called FinLex. As the microblogs are all related to finance, we can create a financial lexicon to depict the sentimental information of the microblogs. We use 2.6 million microblogs (same as the LDA training data) for selecting candidate words. The term frequency, position and part of speech tags are used as features. We totally get 32296 candidate words in this processing. In the next step, we select 40 words in both positive and negative sentiment as the seed words. Then we compute the so-pmi of each candidate words to determine whether the word is positive or negative. For each two sentiments, we conduct the following processing for words separately. We scan the dataset, and draw a line between the two words which are co-occurred in a sentence. After scanning the set, we construct a word relation graph, then we initialize all words' score as 1.0, and run the SentiRank (similar as the PageRank, while the elements of SentiRank are words) to get the emotion score of each words. Then we scale these scores, and make sure they all range between [0, 1]. At last, we drop some words whose score is very low.

In this method, we also consider the influences of some specific adverbs such as "稍微" (slightly), "非常" (very), "否" (not) and so on. We manually create a dictionary and rank all words in it. The dictionary contains 120 degree and privative adverbs. The range of scores that degree adverbs gained is [0.2, 2], while all the scores of the privative are -1.0, because they just reverse the sentiment. We consider the adverbs only if the adverbs appear before the sentiment words, and they are in a same block (a character's sequence between two punctuations is defined as a block).

When the lexicon is constructed, we can compute the positive probability of the microblog using the following formula:

$$\gamma = \Sigma w_i * s_i \quad (1)$$

While the γ is the sentiment (positive or negative) score of the microblog, w_i represents the adverbs weight which appear before the i^{th} sentimental words, s_i is the positive or negative score of the i^{th} sentimental word. According to (1), we can compute the probability of positive (s is positive score) or negative (s is negative score).

2) Binary classification: We also propose a method called BinCla which uses the statistical method to analysis sentiment. First, we conduct a feature selection processing for microblog text representation. Using 2.6 million microblogs, we select

14549 feature words as our dictionary based on TF-IDF and part of speech tags. All microblogs have one or more features words. Second, we use the dictionary to create VSM representation (described before) to depict these microblogs. Third we compute the text vector which represents text by a semantic vector based on Google's word2vec [15]. Fourth we combine the VSM and text vector as the representation of the microblog. Finally, we use SVM as our classification model to predict the sentiments of these microblogs.

3) Combined Method: Finally, based on two facts: 1) the field we work in is finance, it is easy to describe the text by domain lexicon. Thus we propose the FinLex method. 2) The texts we analyze are microblogs, which are almost oral language and free style presented. Thus we propose the BinCla method. Lexicon is difficult to cover all situations, but statistical method can suit this case. Then we combine the method FinLex and BinCla named Comb. The domain knowledge and expression of the microblog are both considered in method Comb. The formula of method Comb is shown as follows:

$$\delta = k_\alpha * a + k_b * b \tag{2}$$

Note that, the δ is the sentiment score computed by Comb, k_a is the weight of method FinLex, and a is the output of method FinLex, and k_b, b is the same in method BinCla. Where $0 <= k_a, k_b <= 1$ and $k_a + k_b = 1$.

4) Sentiment Index: To analyze the stock market by the day level, we generate some aggregating sentiment indexes for each user. We define two indexes, Pos-Neg introduced by [9] [10], positive index proposed by [8].

$$\text{Pos-Neg} = \ln [1+M^{pos}] - \ln [1+M^{neg}] \tag{3}$$

$$\text{Positive Index} = M^{pos} / M^{total} \tag{4}$$

M^{pos} is the total positive microblogs, M^{neg} is the total negative microblogs, and M^{total} is the total microblogs for a user in one day. The Pos-Neg index reflects the comparison between bullishness and bearishness, while the positive index presents the user's bullish prospect of stock market.

D. Stock Prediction

In this module, we combine the stock historical data and the result index of sentiment analysis from previous SA module to predict stock trend of next day. For further improving the prediction accuracy, we also propose a user-group model, which takes full advantage of users' information. We classify users into two categories: star and crowd. A star means an expert and an important poster-generated user. While the number of star is small. The posters of crowd have a little relation with stock market, but the number of crowd is large. Different users have different treatments. Each star has a personal model to be handled, while crowd are analyzed together. As for the computing of sentiment index, each star has a personal index (only use their own posters to compute the index), but for crowd, all crowd users are treat as one user, and use their all posters to compute the index.

The first issue of the SP module is how to select features to predict stock market movement. In [5] [6] [8] [9], all these works prove that microblog sentiments have certain influences on stock market, and the work in [8] suggest that the influences are continuous, and formed a hump-shaped curve. They also state that microblog sentiment lagged for 3-6 days is the best match to stock market variation. Guided by these works, we use sentiment as important information to predict stock market movement. After the SA module' processing, we have already gotten the sentiment indexes, so we can use these indexes as sentiment features to analyze stock market change. In the SP module, we take the sentiment indexes, the past 6 days' stock volumes, the change amount of stock market and the stock index as features to predict stock market movement.

Once the features have been defined, the next important problem is how to create an effective model to predict the movement of stock index. We all know that the microblogs posted by different users have different influences to stock market. According to this fact, we develop our user-group model. We classify users into two categories: star and crowd. The user-group classification depends on the correlation between the users' microblogs and the stock market. We train LR (Logistic Regression) model for each of the users depending on the historical data, and evaluate the accuracy of these models, then select the top accuracy users as the star, others will be crowd. As mentioned earlier, for star users, we remain the personalized models (LR model) for each of them, but for crowd, a unified model (SVM) is developed for all of them. Although the users are divided into two categories, we must combine their performances, and give a certain outcome. Suppose that π is outcome of the combination model, LR_i is result of the i^{th} star user, while p_i is the weight of LR_i, SVM_{crowd} stands for the prediction of all crowd users. The combination has the following form:

$$\pi = \Sigma p_i * LR_i + SVM_{crowd} \tag{5}$$

Since the environment is dynamic, some star users may disappear or may post significantly lower number of related microblogs. In addition, some new users may come up to the star. Thus the constant user-group classification may be inadequate. We address this issue by conducting the classification after a period of time. Reallocating the roles of the users can fit the dynamic changes of the data, and make our model more effective. We call this model as a user-group model. It takes full account of information from the star users and also accepts the voices of the crowd. This means that all users can have effects on stock market in our model. The research of the SSE (Shanghai Stock Exchange) also supports our user-group model [15].

According to the method described above, our user-group model relieves the deviation of the SA module and also tolerates some noise from the MF module. Since the sentiment scores are gained by the SA module, so if a user is unfriendly to the SA module, it will be distinguished as crowd and unified

with others. This will reduce the influence of the SA module's deviation. As for noise, we may note that the posters from star users are mostly financial, because if a user contains noisy posters, its accuracy must be lower and the probability of being a star may be also lower. We just analyze the crowd use the unified model, thus the noise can be drowned by lots of crowds' posters. Although the model can tolerate noise, too much noise will also disturb our model, so the MF module is necessary.

To sum up, our method consists of the MF module, the SA module and the SP module to predict stock market movement. In the MF module, we conduct a method based on LDA and keywords to clean the raw data. In the MF module, we mainly consider the recall of the method, because the method we design is tolerable for noise, but sensitive to losing useful information. The SA module extracts sentiments from financial microblogs obtained in the MF module. We create a financial sentiment lexicon and develop a classification method base on feature selection and word embedding to analyze the sentiment. The measure we mostly concerned in the SA module is accuracy, and we also expect that the negative expression can be effectively distinguished by the SA module, because the negativity is more related to stock market than positivity. The SP module plays an important role in our method, and determines the final performance. It receives the outputs of the SA module and combines the sentiments with historical stock data to predict the stock market movement. Whether the stock market movement is up or down, we can benefit a lot, if we can see it beforehand. So the accuracy of the SP module is the most important indicator we concern.

III. EXPERIMENTS

A. Microblog Filter

We randomly sample 3000 microblogs from our dataset as the test data, and manually label them. All the data are not from the dataset which is used to train the LDA model, but the dataset which is used to analyze the stock market. We conduct the experiment with the KeyWords method, the LDA-3 method (described in subsection B of section II) and the LDA-simple method (similar to LDA-3 but do not extent with keywords). *Table I* displays the result of the three approaches.

TABLE I. MICROBLOG FILTER RESULT

	KeyWords	**LDA-simple**	**LDA-3**
Accuracy	85.47%	85.60%	88.60%
Precision	93.47%	68.26%	67.01%
Recall	33.81%	66.04%	91.04%
F-measure	49.66%	67.13%	77.19%

In this experiment, the 100/200/300/400 topics are set up to test the influence of topic number and to find the optimal one. The result suggests that the performance is almost not improved when the topic number is bigger than 200. We also tested the asymmetric Dirichlet prior [17] and the performance is not improved much compared to symmetric prior. So for simply and effectively, we use 200 topics and the symmetric Dirichlet prior in our method.

As shown in *Table I,* the result indicates that our method is outstanding. The KeyWords contains 79 words which are very relevant to finance, so the precision is high, but it causes a serious problem that the recall is lowest. This means most financial microblogs are discarded by this method. The two LDA methods both have a little low precision, because LDA is a probabilistic model, every word has a probability value in every topic, this may cause that some noisy microblogs which have similar expressions with the finance may be regarded as financial text by LDA, such as some company information and sharing experience speeches of the CEOs. The LDA-simple takes the semantic feature of microblog, so it gets a higher recall, but its precision is lower. Our LDA-3 considers both the semantic feature and keywords, so we can get the highest recall, and a similar precision compared to the LDA-simple. Our LDA-3 guarantees that almost all financial microblogs are retained without much noise included, so it gets the highest F-measure score. This is what we want. It doesn't matter that our data contains a little noise, but it is vital that mostly all information is included. The recall is the main indicator for the MF module, and the precision should be also considered, so the F-measure our evaluation criterion for the MF module. According to our experiment, the LDA-3 is the best method for this mission.

B. Sentiment Analysis

We have randomly selected 4500 microblogs to test the SA module. All of these microblogs are classified into three categories (positive, negative and neutral) by nine students. Each microblog is labelled by three students, and we only use microblogs that are same classified by all three.

In our experiment, we compare our lexicon (Method FinLex) to the NTUSD created by Taiwan University (NTU), which is a common dictionary in Chinese NLP task. The BinCla method is our classification method described in subsection C in section II, and the Comb method is the combination method (combine FinLex and BinCla). The result is shown in *Table II:*

TABLE II. SENTIMENT ANALYSIS RESULT

	TPR	TNR	Accuracy
NTUSD	73.47%	45.70%	62.36%
FinLex	79.93%	49.46%	67.74%
BinCla	81.36%	46.23%	67.31%
Comb	73.03%	60.25%	68.17%

Here we define four indicators represented as follows: *tp* is the number of correctly identified positive microblogs; *tn* is the number of correctly identified negative microblogs; *p* is the number of total positive microblogs in the dataset; while *n* is the number of all negative microblogs in the dataset. The TPR

= *tp / p* and TNR = *tn / n*. TPR reflects the method's identified power for positive microblogs, while TNR reflects the power for negative microblogs.

Obviously, FinLex is better than NTUSD, which means that our lexicon is available for financial microblogs. BinCla has a similar result compared to FinLex, although the BinCla takes advantage of the free language style, polysemy and neologisms, it also loses the domain knowledge. Comb is a combination of FinLex and BinCla and it gets the best performance. One reason for the progressive performance is that FinLex and BinCla deal with the problem in different views, so the Comb can complement one another. Here, we may note that TNR is lower than TPR. The reason is the negative expressions are mostly obscure. Although it hard to distinguish the negative expressions, we must pay more attention to it, as negativity is sensitive to stock market. Comb achieves a progressive performance in TNR with just a little drawback in TPR, while its total precision is also the best. So it is our choice to analyze the sentiment.

C. Stock Prediction

We also test the SP module. The data we used spans September to December in 2015. The microblogs are collected by our extraction system and shared by pennyjob, totally about 6.1 million. The stock data is downloaded from NetEase Money. Here is our result in *table III:*

TABLE III. STOCK PREDICTION RESULT

	Accuracy
LR-Only	53.70%
LR-OneClass	61.11%
SVM-Only	62.96%
SVM-OneClass	64.81%
User-Group Model	69.09%

LR and SVM are the mostly used to analyze stock market. The LR-Only is a model without sentiment, while LR-OneClass contains sentiment but without user-group model, the same as the SVM-Only and SVM-OneClass. User-group model is described in subsection *D* of section II. In comparing only method (LR-Only and SVM- Only) and oneclass method (LR-OneClass and SVM-OneClass), we can declare that the sentiment is useful for stock market prediction. Adding user classification, the performance increases from 64.81% to 69.09%. It is clear that our user-group model is promising and outstanding.

IV. CONCLUSION AND FUTURE WORK

We present a systematical method to predict stock market movement, develop meticulous methods for the MF module and the SP module, create a lexicon for financial microblog, and improve the performance of the SA module. Our MF module benefits from LDA for extending the semantics of microblogs. For the SP module, we propose user-group model, which delicately depicts the information user posted. The experiment results also suggest that our work is well-directed and promising.

In the future, we will consider the one-class classification for the MF module, and an ensemble method for the SA module. We also think about the stream work flow to conduct this research. Also the labeled dataset will be enlarged.

ACKNOWLEDGMENT

We would like to thank pennyjob for the sharing data, and Huaping Zhang for the open NLPIR software. We also thank the reviewers for their valuable comments. This work is funded in part by China NSF Grants (No.61572250 & No.61223003) and Jiangsu Province Industry Support Program(BE2014131).

REFERENCES

[1] Syed Aqueel Haider and Rishabh Mehrotra. Sentiment polarity identification in financial news: A cohesionbased approach. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp.984–991.

[2] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using Structured Events to Predict Stock Price Movement:An Empirical Investigation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp.1415–1425.

[3] Christopher C. Chua, Maria Milosavljevic, and James R. Curran. A Sentiment Detection Engine for Internet Stock Message Boards. In Proceedings of the Australasian Language Technology Association Workshop, 2009, pp. 89–93.

[4] Werner Antweiler, and Murray Z. Frank. Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. The Journal of Finance, 2004, 59(3), pp. 1259-1294

[5] Roy Bar-Haim, Elad Dinur, Ronen Feldman, MosheFresko, and Guy Goldstein. Identifying and following expert investors in stock microblogs. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2011, pp. 1310–1319.

[6] Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. Exploiting Social Relations and Sentiment for Stock Prediction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1139–1145.

[7] J. Bollen and H. Mao. Twitter mood predicts the stock market. Journal of Computational Science, 2011, pp. 1-8.

[8] Malcolm Baker and Jeffrey Wurgler. Investor Sentiment in the Stock Market. Journal of Economic Perspectives, 2007, 21(2), pp. 129-152.

[9] Marta Arias, Argimiro Arratia and Ramon Xuriguera. Forecasting with twitter data. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 5(1), pp. 8:1-8:24.

[10] Chong Oh, and Olivia Sheng. Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement. Thirty Second International Conference on Information Systems (ICIS), 2011, Paper 17.

[11] David M. Blei, Andrew Y. Ng and Michael I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research. 2003, 3(1) pp. 993-1022.

[12] G. Salton, A. Wong, C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 1975,18(11) pp. 613-620

[13] Eduardo J. Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Correlating Financial Time Series with Micro-Blogging Activity. In Proceedings of the fifth ACM international conference on Web search and data mining (WSDM), 2012, pp. 513-522.

[14] M. Ghiassi, J. Skinner, D. Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural

network. Expert Systems with Applications (ESWA), 2013, pp. 6266–6282.

[15] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, 2013. arXiv:1301.3781, 2013

[16] http://www.sse.com.cn/aboutus/research/jointresearch/c/plan20020412r.pdf

[17] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking LDA: Why priors matter. In NIPS 22, 2009.