2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)

Stock Prediction Using Event-based Sentiment Analysis

Masoud Makrehchi
Department of Electrical, Computer, and Software Engineering
University of Ontario Institute of Technology
Oshawa, ON, Canada
Email: masoud.makrehchi@uoit.ca

Sameena Shah[1] and Wenhui Liao[2]
Research and Development, Thomson Reuters
[1]New York, NY, USA
[2]Eagan, MN, USA
Email: {sameena.shah,wenhui.liao}@thomsonreuters.com

***Abstract*—We propose a novel approach to label social media text using significant stock market events (big losses or gains). Since stock events are easily quantifiable using returns from indices or individual stocks, they provide meaningful and automated labels. We extract significant stock movements and collect appropriate pre, post and contemporaneous text from social media sources (for example, tweets from twitter). Subsequently, we assign the respective label (positive or negative) for each tweet. We train a model on this collected set and make predictions for labels of future tweets. We aggregate the net sentiment per each day (amongst other metrics) and show that it holds significant predictive power for subsequent stock market movement. We create successful trading strategies based on this system and find significant returns over other baseline methods.**

I. Introduction

A well-known behavioral economics hypothesis states that public mood and market performance are correlated. The idea being that when people are happy, optimistic, and in a good mood, they are more likely to increase investment, which in turn improves stock market performance. However, quantification of public mood is not a trivial task. In order to build a brute force predictive model that quantifies public mood and predicts future market performance, we need to have access to every single individual's mood and aggregate all moods to estimate the public or collective mood score. Since observing every person's mood is not possible, a very small sample of the population may be used.

Social media (especially Twitter) can provide a sizable population sample and also access to user generated and shared publicly available content. This content may be mined to learn sentiment associated with public mood.

In this paper, we use two approaches for mood and sentiment detection. The first approach called lexicon-based mood detection, is usually considered as an unsupervised learning approach. The lexicon-based approach involves calculating presence and counts of chosen words or phrases in the document. The second approach, which is also our proposed method, is based on supervised learning. Generally supervised learning gives better results on most problems but is impractical for certain use cases because of lack of labeled data. This is especially true for tweet data. We propose a solution to avoid this problem by extracting automated labels from the stock market based on good (for example, significant increase) or bad days. In our proposed approach, we monitor market performance and find out which days are good or bad days for the market. One of the main contributions of this research is to build a training data set for supervised learning of sentiment based on these significant events, that is, the times that witness a sharp decline or rise in market performance. Using the collected training data containing tweets with positive and negative sentiments, we are able to build a classifier and predict sentiments of tweets on the fly. Aggregating the predicted sentiments gives us a sentiment score. Our study shows that the aggregated sentiment score has a decent correlation with the market performance.

In general, our technique of assigning labels may be used for any quantifiable metric like stock price movement, oil prices, earnings announcements, or other macro-economic factors like job numbers etc. The labels need not be restricted to binary, and may be discrete. The time frame for computation of returns can range from intra-day to daily to yearly. The text that needs to be labeled can be either pre-event, post event, or contemporaneous according to the use case.

The paper consists of six sections. Following the introduction, sentiment analysis and public mood detection are briefly reviewed in Section II. The event-based training data generation for sentiment analysis is discussed in Section III. Section IV provides details for the proposed approach and the implementation the supervised model. Experimental results and conclusions are presented in Sections V and VI respectively.

II. Related work

The main motivation in sentiment analysis is to find out what other people think[1]. Although sentiment analysis of narrative text is a mature field, learning sentiment from web documents was initiated about a decade ago. Especially in recent years there is a marked increase in sentiment analysis on social media such as Twitter and Facebook.

O'Connor et al. [2] study relations between tweet sentiment and public opinion. Public opinions are observed by surveys and polling systems. Wilson et al. [3] measure tweet sentiment as the ratio of positive and negative words in a tweet obtained using OpinionFinder [4].

978-1-4799-2902-3/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.48

Asur et al. [5] focus on predicting box office revenue of movies using sentiment from twitter messages. Their main hypothesis was that the more a movie is talked about, the more likely it is to be successful at the box office. They performed prediction using the following two approaches. (i) Without any sentiment analysis: A mention in a tweet is assumed positive. The underlying assumption is that people post positive comments about movies way more than they post negative comments. They use this to construct a feature called "tweet-rate" and use a regression model to perform prediction.(ii) With sentiment analysis using an N-gram language model. Gold data was generated using Amazon Mechanical Turk and the prediction was compared with HSX (Hollywood Stock Exchange). Their overall results showed that twitter data can be used for predicting box office success.

Bandari et al. [6] propose a method to predict the popularity of news items on the social web. For each news article, features are generated based on the source of the news story, news category, subjectivity of the news, and named entities mentioned in the news. They apply regression and classification algorithms to predict news popularity based on these features. Their experiments showed that it is possible to estimate ranges of popularity with an overall accuracy of 84% considering only content features.

In contrast to [6], Hong et al. [7] predict popularity of recent messages on twitter. They model the problem as a classification problem, and construct features based on message content, temporal information, metadata of messages and users, as well as the users' social graph.

Gilbert et al. [8] estimate anxiety, worry and fear from over 20 million posts on LiveJournal, and find that an increase in these negative expressions predict downward pressure on the S&P 500 index. Zhang et al. [9] describe a simple approach to predict stock market indicators such as Dow Jones, NASDAQ, and S&P 500 by analyzing twitter posts. The authors estimate tweet mood based on tweet count of words expressing certain pre-decided mood (hope, fear, worry), number of followers of the mood, number of re-tweets of the moods, etc.

Bollen et al. [10] also uses twitter mood to predict the stock market. The sentiment analysis is based on OpinionFinder [4] and POMS [11]. OpinionFinder assigns a positive or negative polarity to a tweet, while POMS assigns one of the following six labels: calm, alert, sure, vital, kind, and happy. A time series of mood is constructed using collective tweet sentiment per day. Their analysis shows strong correlation between 'calm' mood and DJIA data while others including OpinionFinder sentiment shows weak correlation.

Oh et al. [12] forecast future stock price movement with micro blog postings from StockTwits and Yahoo Finance. They constructed multiple sentiment classifiers based on posts that were manually labeled with bullish, bearish, or neutral sentiment. The sentiment scores were used to predict the future direction of the stock market. Ruiz et al. [13] use tweets about specific stocks and represent tweets through graphs that capture different aspects of the conversation about those stocks. Two groups of features are then defined based on these graphs: activity-based and graph-based features. By studying the relationships between these features and the traded volume and price of stocks, they develop a trading strategy that outperforms other baseline strategies.

In all these approaches for sentiment analysis, either an unsupervised method, such as a list of predefined mood words is used, or training data are manually generated. We generate training data automatically to build a sentiment classifier. One big advantage of this method is that we can label as much as data we need, and no human labeling efforts are needed at all.

With regards to our event-based supervised learning method, we are not aware of any similar effort in assigning labels to social media text. Perhaps, the closest system to our work is that of Lavrenko et al. [14] in which stock trends were used to label selected news stories. The authors chose news stories that were known to be related to each stock. We concentrate our efforts on social media text rather than news documents. From a linguistics perspective, social media text (also called micro-text) is considered inherently different from other types of documents. Social media text has much more noise and lesser content in each document (or tweet). Also, the linguistic quality and use of some special notations like hashtags, RE, re-tweet, etc. render additional differences. Social media text is much more representative of the general public in terms of both opinions as well as linguistic quality than those witnessed in news articles written by journalists. While the amount of content per social media documents might be very less, the scale of social media text is enormous and continually increasing. This makes it impossible to manually curate training data, hence our system is even more important in this regard.

There are two aspects to the proposed method, (i) learning sentiment from relevant labeled text using significant events, and (ii) using the learnt sentiment to predict future significant events. We discuss learning of sentiment in the next section, followed by prediction in section IV.

III. Supervised Sentiment Analysis

As discussed in the previous sections, our method takes advantage of supervised learning algorithms while avoiding manually labeling training data. There are two key components in learning sentiment: (i) automatically generating training data from unlabeled user tweets; (ii) building a sentiment classifier based on the training data.

A. Automated Generation of Sentiment Training Data

Given a tweet, in order to predict its sentiment, either unsupervised or supervised learning may be used. The benefits of using unsupervised learning is that no human-tagged training data is needed, however, it usually doesn't perform as well as supervised learning. For supervised learning, various classification techniques can be used to model tweet sentiment. However, it is usually very costly if not impossible, to get human-tagged training data, especially for data the scale of tweet data.

Since in this paper our goal is to predict stock markets, we consider only stock related sentiments. In general, big stock price fluctuations affect people's mood, and people's mood in turn affects stock market. We therefore assume that when a stock price goes significantly up relative to overall stock market (for example, S&P 500), in general, the tweets related with this stock reflect positive mood; similarly when a stock price goes way down with respect to S&P, the related tweets reflect negative mood. Therefore, we can use the stock price changes to label a tweet's mood.

Apart from using stock returns, other explicit positive and negative events may be used to label tweets. Corporate Earnings reports are one such type of events that may cause significant stock price movement. When a company beats consensus estimation of its earning and revenue by a big margin, the sentiment towards the company usually turns positive; vise versa. Since historical data of how much a company beats expectations is available, we can use those events to label tweets. Other events such as merger-and-acquisition announcements, macro-economic events such as monthly job reports of US unemployment rate, monetary policy announcements from central banks around the world, can all influence stock markets dramatically, and thus can be used to label our data.

Apart from these, some unexpected events can change people's mood dramatically. For example, when Jamie Dimon, CEO of JP Morgan, first announced the two billion loss caused by the "London Whale" trades, the sentiment towards the company turned extremely negative in the next 48 hours. The technical problems that caused the delay of Facebook IPO played a big role in affecting people's sentiment towards Facebook stock and IPO in general.

When we label a tweet based on the clues mentioned before, a positive label means at the moment when the user posts the tweet, the sentiment is positive. We could label tweets with temporal sentiment, such as whether a user expects, experiences, or recalls positive/negative sentiment. We could also use different time frames for computation of returns, for example, intra-day, daily, monthly, or yearly. Depending on the use case, the assigned text can be either text that was generated pre-event, contemporaneously, or post-event.

In the experiments reported in this paper, we label each tweet based on daily events. Let t_o and t_c be the opening and closing time of a particular stock, or index, S_i. Here, $t_c - t_o = 8hrs$ are the official market operation hours. Let r_{sp} be the return for S&P 500 and $F(t_o)$ and $F(t_c)$ be the opening and closing price of the stock. We mark a period of time as a gain event if

$$(F(t_c) - F(t_o))/F(t_o) > 3.0\% + r_{sp} \ . \tag{1}$$

Similarly, we mark a period of time as a loss event if

$$(F(t_c) - F(t_o))/F(t_o) < -3.0\% + r_{sp} \ . \tag{2}$$

Other market events that do not meet these two criteria are ignored. When observing a gain event, all tweets mentioning the stock S_i on that day are collected and labeled as positive tweets. Similarly, all tweets collected during a loss event are labeled as negative tweets. If a stock has more than one gain or loss event, all of them are considered. Using this process, a set of positive and negative examples are generated and we are able to implement a supervised model to learn about the sentiment of unknown tweets.

B. *Sentiment Classifiers*

Having both positive and negative examples, we implement two types of classifiers. The first classifier uses heuristic features from our mood word list such as "happy", "hope", "sad", and "disappointed". We use an extended list of mood words that includes over one hundred mood words and their derivations. Every tweet is represented by a feature vector with the same size of mood word list. If mood word i in the list occurs in the tweet, the corresponding feature in the vector turns on (one) otherwise it will be off (zero).

For the second classifier, a feature engineering task is performed to select positive features associated with positive tweets and negative features associated with negative tweets. In many cases, training data is unbalanced (many positive examples vs. few negative examples and vice versa). To deal with the class imbalance problem, an over sampling technique at feature level instead of data is employed. In other words, if the number of positive examples is 50% of negative data, in the final feature vector, the number of positive features will be two times bigger than the number of negative examples. With this strategy, we let minorities have more representative data in the training process.

In addition to the positive/negative features, we also use features based on the metadata associated with each tweet, including the existence of hashtags, stock ticker symbols, URLs, and re-tweets (RT), etc. However, we did not find any significant predictive power in these features.

We call the classifier using the mood word list as the baseline classifier and the second one is called the sentiment classifier. Although the positive and negative features are not necessarily sentiment words (such as the word "treasury"), in combination the words express a sentiment toward a gain or loss event in the market.

For both feature vectors, a Rocchio classifier[15], which is simple yet effective in text classification tasks, is employed. Both classifiers offer precision and recall over 90%. Sentiment classifier offers slightly better results as given in Table I. The precision and recall are estimated using 10-fold cross validation over training data.

However, the true test is to realistically evaluate how indicative the approach is in predicting future stock price direction, the eventual aim of the process. In section IV, we provide experiment results for these two approaches and test how they fair individually and against each other.

IV. Prediction Algorithm

In the previous sections, we discussed the learning of sentiment using significant events. In this section, we discuss

Algorithm 1 Stock market prediction using event-based supervised sentiment learning

Choose the significant event criteria.
Choose appropriate pre/post/contemporaneous tweets based on the chosen significant event criteria.
Assign an appropriate label for each tweet based on its associated event (for example, positive for rise or negative for fall).
Train a classifier on the labeled tweets.
Predict sentiment of new future tweets.
Aggregate tweet sentiments.
Take a long/short position based on the net aggregated sentiment.

prediction of sentiment for future tweets. Below, we summarize the necessary steps in the proposed algorithm:

As mentioned previously, significant events can be defined based on the specific use case. We chose significant stock movement compared to the S&P 500 as the event criterion. For the training data, text/tweets that were generated pre-event/post-event/comtemporaneously may be collected and assigned labels. In the results reported in this paper, we collected and used all of them. We assigned binary labels, positive or negative, to tweets. Labeling need not be restricted to a binary set, any granularity or number of classes may be used. Also, the problem need not be modeled as a classification problem, regression may be used. The underlying model should be used to predict the label (or sentiment) of a future tweet. The labels are subsequently aggregated and then used to make a prediction for stock market. The idea being that the system would learn the sentiment of tweets before significant positive and negative events. Once this sentiment (or pattern in general) is observed again in the tweets, it would be indicative of future stock market performance.

A. Descriptive Statistics about the Datasets

We collected tweets for all stock constituents of DJI30 index from 27th March to 13th July 2012 using Twitter's Search API. The Search API returns all recent tweets that include the search query. We manually curated a query list in which we included all the terms that are used to refer to a company on Twitter. For example, for Apple Inc. our search query was "Apple Inc OR AAPL OR #AAPL OR $AAPL OR AAPL". This may not be foolproof, but the idea is to look at aggregated sentiment from tweets and hence some noise should not have a significant impact on the results. Since the Search API only returns a recent history, we ran a cron job every 15 minutes that iteratively searched and stored all unique tweets in a database. Tweets returned by the Search API have unique identifiers and hence are easily distinguishable. This led to a total collection of about 30 million tweets. The distribution of tweets across companies is not uniform with some companies being far more popular than others in the index. For example, 'Apple Inc.' is far more popular than 'The Travelers Company'.

After training on the tweets that satisfies the significant event criteria according to equations (1) and (2), we obtained predictions for about 2 million tweets. The classifier also generated a confidence score for each predicted label. The confidence probability across all tweets ranged from 0 to 1 with a median of 1 and mean of 0.9472. We eliminated tweets with a confidence score of less than 0.80 which removed about 7% of tweets. We found that the removal of this set does not have any significant impact on the results.

TABLE I
AVERAGE PRECISION, RECALL, AND F-MEASURE OF THE SENTIMENT CLASSIFIER.

	Mood words (Baseline)	Proposed approach
Precision	0.83	0.91
Recall	1.00	1.00
F-measure	0.91	0.96

The proposed approach for generating training data for sentiment analysis is evaluated using two methods: internal and external. The internal method employs a traditional classifier evaluation approach. A good classification accuracy shows that labels have consistent distinguishing textual patterns learnt by the classifier. This is indeed necessary but not sufficient. The external evaluation tests whether these have predictive power or not. The predicted labels should be able to predict the events that were used for initial labeling. If reasonable predictive power is seen, then this can be exploited to learn characteristics of any event of interest, in our case stock market performance.

B. Internal Evaluation of Sentiment Classifier

Similar to traditional classifier evaluation procedure, the labeled tweets are divided into training and test data. The classifier is trained on training data and evaluated on test data. Using 20 times 10-fold cross validation, average precision, recall, and F-measure are estimated. Table I illustrates the result of internal evaluation, which is quite promising.

As seen both classifiers have reasonable accuracies, with the sentiment classifier outperforming the baseline classifier. Next, we perform an external evaluation to test for predictive power.

V. EXPERIMENT RESULTS FOR STOCK MARKET PREDICTION

In this section, we give our results for testing the aggregated sentiment as a predictor of the S&P 500 index. We obtained the daily prices for S&P 500 and computed daily returns as $close/open - 1$. Since we want to predict next day's performance, we aligned tweets with the next trading day's returns. We adjusted the trading day for weekends and other holidays.

A. Quality of feature

We begin by testing the stock performance predictive quality of sentiment as a feature. Such a test can be performed using the standard bucketing procedure. We computed mean positive and mean negative sentiment for each day. We computed net sentiment as $PositiveSentiment - NegativeSentiment$

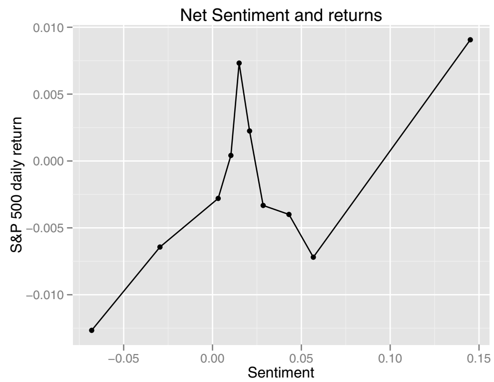

Fig. 1. This graph shows the mean returns obtained per net sentiment decile. As seen, when net sentiment is extremely negative, the returns are lesser and for extreme positive, the net returns are significantly positive.

and divided it into deciles. Figure 1 shows the mean return obtained for its respective sentiment decile. As seen, when net sentiment is extremely negative, the returns are lesser and for extreme positive, the net returns are significantly positive. This builds the intuition that the feature certainly holds predictive power around extreme sentiment regions. Around the net zero sentiment (or neutral) the returns are not very indicative because there are other factors that might be at play. This is inline with our expectations.

One key issue that arises is that of aggregation of tweet sentiment. Tweet sentiment can be aggregated using different functions as well as different filters. We tested aggregated sentiment as a predictor for the S&P 500 index. We also tested aggregated sentiment as a predictor of individual stock performance.

B. *Aggregation over entire day*

Figure 2 shows the results for a simple directional trading strategy that takes a long position on the index if the previous day's sentiment is net positive and a short position if the previous day's net sentiment is negative. The results show that both the approaches are good predictors of next day's direction. Both the approaches are more profitable than holding the index. However, the supervised learning approach generates significantly higher returns over the lexicon based approach. Hence it is clear that there is some predictive power in the learnt sentiment.

Based on our different data analysis steps, we observed that the twitter sentiment time series has autocorrelation, which seems reasonable considering human emotions may be dependent on recent past human emotions. We created the following autoregressive model to account for this historical dependence.

$$\begin{aligned} SPX_{return}(1) \sim class(-1) + class(-2) + class(-3) \\ + pos(0) + neg(0) + class(0) + SPX_{return}(-1) \\ + SPX_{return}(-2) - 1 \end{aligned} \quad (3)$$

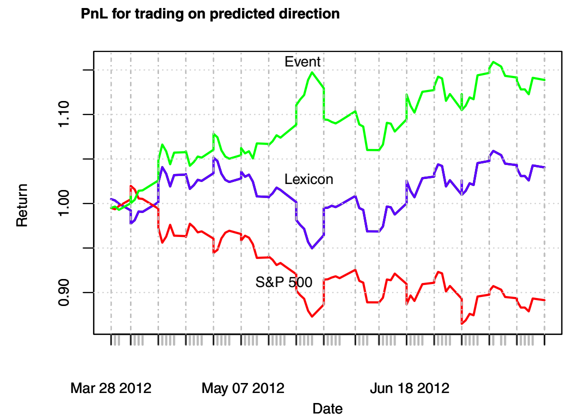

Fig. 2. Pnl (Profit and Loss) when the trading strategy was to take a position determined by previous day's net sentiment. If previous day's net sentiment was positive, we went long the SPY 500 the next trading day (adjusted for holidays) and vise versa. As seen in the graphs, while both strategies are good predictors of the S&P direction, the supervised learning approach outperforms the lexicon approach.

In equation (3), $SPX_{return}(t)$ represents the daily arithmetic return, defined as (open/close -1), of the S&P 500 on day t. The regression is run daily, where the day is denoted as day 0. $t = -1$ implies 1 trading day before day 0. $pos(t)$ denotes daily aggregated positivity of the twitter series on day t. $neg(t)$ denotes daily aggregated negativity of the series, and $class(t)$ denotes $sign(pos(t) - neg(t))$ on day t. The constant factor in the above is used for removing intercept. On each day, a training set of last ten days of such data is calculated and regressed upon. A prediction is then using the fit obtained for the next days return. Figure 3 shows the results. As seen in the graphs, both the strategies give better performance than a simple one day strategy, and the proposed method performs significantly better than the lexicon-based approach. Of course both strategies are better than holding the SPX even after reasonable trading costs may be removed. Hence, training on recent historical data gave us better returns. Note that the trading strategy based on the proposed method gives about 15% returns in about a four month period.

1) Individual stock: In the second experiment, we aggregated sentiment per stock per day and went long the next day on the stocks with sentiment more positive than the moving mean sentiment. Similarly, on the next trading day, we shorted the stocks with sentiment lesser than the moving mean sentiment. Figure 4 shows the returns of both the event-based and lexicon-based approach. Both approaches outperform the index, but as seen the event-based approach has more consistent returns and a higher Sharpe ratio. When two strategies give similar performance, the strategy with the higher Sharpe ratio is much more favorable because of the lower risk (or variability in returns) incurred[1].

[1]It is easy to tell that the event based method has a significantly higher Sharpe ratio because of the consistent uptrend.

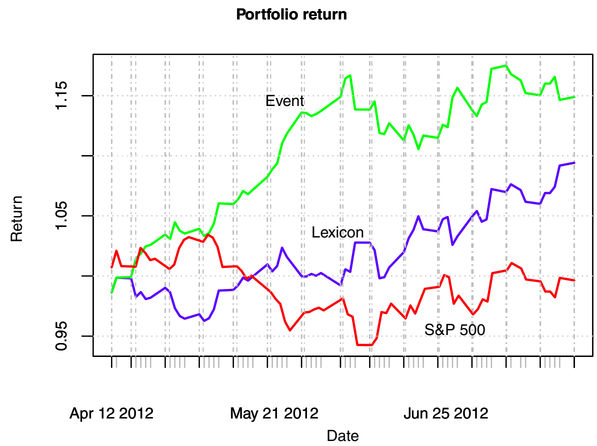

Fig. 3. Pnl when the trading strategy used autoregressive features and trained on last ten days features. The proposed supervised learning approach outperforms the lexicon approach. Both strategies significantly outperform S&P 500 significantly.

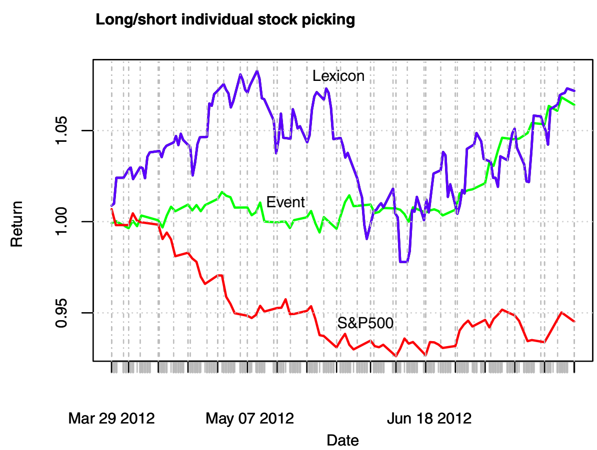

Fig. 4. Pnl when the trading strategy went long on stocks with sentiment greater than moving mean sentiment and short otherwise. Both strategies significantly outperform S&P 500 significantly, with the event based approach showing a higher Sharpe ratio.

VI. Conclusion and Future Work

We proposed a novel method for estimating sentiment based on twitter posts and use the sentiment to predict future stock market movement. Specifically, we automatically generate training data based on events related with stock markets. With such training data, we are able to build a high-precision and efficient classifier to assess tweet sentiment and use such information to build an effective trading strategy. Our trading strategy is able to beat S&P 500 by about 20% returns in four months.

The fundamental point is that we use obvious clues (for example, significant market events) that can be easily and automatically detected to get enough training data. Then we build a model to learn patterns from such data, in order to identify obscure clues (for example, hidden factors that affect stock prices). Thus, our method is flexible enough to be used in other applications.

However, there is still room to improve the precision, for example, by employing more sophisticated classifiers, which is one of our objectives in future work. For an external evaluation, the ideal approach would be a domain transfer test using test data from a different domain, for example, movie rating or product reviews. However, this needs labeled test data which was not available in this research. However, we will investigate this approach as well in our future work.

Acknowledgment

The authors wish to thank Hemali Majithia, Thomson Reuters R&D for her support in collecting the tweets. This work was funded by Thomson Reuters Global Resources.

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls : Linking text sentiment to public opinion time series," 2010.

[3] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), pp. 347–354, Association for Computational Linguistics, 2005.

[4] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, "Opinionfinder: a system for subjectivity analysis," in *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, (Stroudsburg, PA, USA), pp. 34–35, Association for Computational Linguistics, 2005.

[5] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, (Washington, DC, USA), pp. 492–499, IEEE Computer Society, 2010.

[6] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," *CoRR*, 2012.

[7] L. Hong, O. Dan, and B. D. Davison, "Predicting popular messages in twitter," in *Proceedings of the 20th international conference companion on World wide web*, WWW '11, (New York, NY, USA), pp. 57–58, 2011.

[8] E. Gilbert and K. Karahalios, "Widespread worry and the stock market," in *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[9] X. Zhang, H. Fuehres, and P. A. Gloor, "Predicting stock market indicators through twitter "I hope it is not as bad as I fear," in *The 2nd Collaborative Innovation Networks Conference - COINs2010*, 2010.

[10] J. Bollena, H. Maoa, and X. Zengb, "Twitter mood predicts the stock market," in *Journal of Computational Science*, 2011.

[11] D. McNair, J. P. Heuchert, and E. Shilony, "Profile of mood states," in *Bibliography*, pp. 1964–2002, 2003.

[12] C. Oh and O. Sheng, "Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement," in *ICIS 2011 Proceedings*, pp. 57–58, 2011.

[13] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pp. 513–522, 2012.

[14] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Mining of concurrent text and time series," in *KDD*, 2000.

[15] J. Rocchio, *Relevance Feedback in Information Retrieval.* Prentice Hall, 1971.