An Experiment in Integrating Sentiment Features for Tech Stock Prediction in Twitter

Tien Thanh Vu[1,3] *Shu Chang*[2,3] *Quang Thuy Ha*[1] *Nigel Collier*[3]
(1) University of Engineering and Technology, Vietnam National University Hanoi, 144 Xuanthuy street, Caugiay district, Hanoi, Vietnam
(2) University of Bristol, Senate House, Tyndall Avenue, Bristol BS8 1TH, UK
(3) National Institute of Informatics, National Center of Sciences Building 2-1-2 Hitotusbashi, Chiyoda-ku, Tokyo 101-8430, Japan
tienthanh_dhcn@coltech.vnu.vn, shuchang0011@gmail.com, thuyhq@vnu.edu.vn, collier@nii.ac.jp

ABSTRACT
Economic analysis indicates a relationship between consumer sentiment and stock price movements. In this study we harness features from Twitter messages to capture public mood related to four Tech companies for predicting the daily up and down price movements of these companies' NASDAQ stocks. We propose a novel model combining features namely positive and negative sentiment, consumer confidence in the product with respect to 'bullish' or 'bearish' lexicon and three previous stock market movement days. The features are employed in a Decision Tree classifier using cross-fold validation to yield accuracies of 82.93%,80.49%, 75.61% and 75.00% in predicting the daily up and down changes of Apple (AAPL), Google (GOOG), Microsoft (MSFT) and Amazon (AMZN) stocks respectively in a 41 market day sample.

KEYWORDS: Stock market prediction, Named entity recognition (NER), Twitter, Sentiment analysis.

Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data, pages 23–38,
COLING 2012, Mumbai, December 2012.

1 Introduction

Recent research into social media has looked at the application of microblogs for predicting the daily rise and fall in stock prices. In many ways microblogs are an ideal early warning about company price movements as they are freely available, rapidly updated and provide spontaneous glimpses into the opinions and sentiments of consumers about their future purchasing behaviors. Previous work such as (Bar-Haim et al., 2011) has assumed that messages containing explicit buy or sell signals about stocks are the most informative for stock price prediction although such messages typically comprise only a fraction of the available information about company sentiment. In this work we approach the task from another perspective, one in which Twitter users' sentiment, related to the company, influences the next day's market price movement. This allows us to tap into a much wider base of mood about the company's future prospects. With the high level of number of Tweets related to tech companies, we focus on predicting stock market movement of four famous tech companies namely Apple, Google, Microsoft and Amazon.

Our approach departs from another major work into sentiment analysis for tracking the Dow Jones Industrial Average (DJIA) by (Bollen et al., 2011b) in that we do not pre-assign a sentiment lexicon or assume mood dimensions. Instead we induce the lexicon automatically by association with "bullish" (a positive price outlook) and "bearish"(a negative price outlook) anchor words on the Web. Further, our work predicts stock market at company level which is deeper than whole stock market level in (Bollen et al., 2011b).

Our work seeks to contribute on several fronts: we explore the underlying relationship between sentiment about the company and stock price movements - helping to avoid the problem of identifying expert stock price pickers (Bar-Haim et al., 2011); we automatically discover sentiment bearing words that have high correlation to the stock market domain; and finally we build a named entity recognition system on Twitter data to identify and remove noise Tweets.Through a series of experiments we show the contribution of sentiment, named entities and changes in the stock market indices on a companies' future share price movement.

Although this was not our initial aim, the methods employed might also offer insights to economists about the causality relation and timing of the response between consumer sentiment and stock price movements which are traditionally seen as being related through expectations about future consumer expenditure.

The rest of paper is organized as follows: in section 2, we provide background. We describe our method and experiments in section 3 and section 4 respectively. The conclusion and future works will be presented in section 5.

2 Background

In recent years many techniques have been applied to sentiment analysis for knowledge discovery in different domains. In early work on product recommendations (Turney, 2002) made use of an unsupervised learning approach to measure sentiment orientation for 410 Epinions reviews using adjectives and adverbs with respect to two anchor words, "excellent" and "poor". His method achieved an average 74% accuracy on four different review domains. (Pang et al., 2002) applied three supervised learners for classifying bidirectional sentiment in movie reviews, finding a number of challenges over traditional topic classification such as "thwarted expectations". (Hu and Liu, 2004) and (Dave et al., 2003) also performed product classification with the former focusing on the qualities of product features. (Mullen and Collier,

2004) used lexical clues from Epinion movie reviews and Pitchfork Media music reviews, yielding insights into the value of topical references.

With respect to identifying subjectivity, (Hatzivassiloglou and Wiebe, 2000)(Wiebe et al., 2001) examined the role of adjective classes for separating subjective from objective language.

The technologies used for determining sentiment orientation commonly include manual or semi-automatic methods for constructing sentiment lexicons, e.g. (Turney, 2002). (Das and Chen, 2007) in the stock analysis domain used a lexicon of finance words to help determine significant correlation between aggregated stock board messages by small investors and the Morgan Stanley High technology 35 Index (MSH35). However, they found the correlation was weak for individual stocks.

More recently, with the explosion of interest in social networks, a popular microblogging service called Twitter has become a major source for data-driven investigation. (Java et al., 2007)(Kwak et al., 2010) for example showed the social motivations of its users, and others (Zhao et al., 2007)(Lin et al., 2010)(Ritterman et al., 2009)(Petrović et al., 2010)(Petrovic et al., 2012) focused on breaking news or event detection. Sentiment analysis has been found to play an significant role in many applications (Krishnamurthy et al., 2008)(Bollen et al., 2011a)(Kivran-Swaine and Naaman, 2011) complementing evidence from Twitter messages and network structure.

In recent work on stock market prediction, (Bar-Haim et al., 2011) used Twitter messages (*Tweets*) from StockTwits to identify expert investors for predicting stock price rises. They used a support vector machine (SVM) to classify each stock related message to two polarities - "bullish" and "bearish" and then identified experts according to their success. The authors found that an unsupervised approach for identifying experts and combining their judgments achieved significantly higher precision than a random baseline, particularly for smaller numbers of experts. However, predictive performance was still low.(Bollen et al., 2011b) employed a set of expression patterns to extract opinions and then map those features into six sentiment orientations, "Calm", "Alert", "Sure", "Vital", "Kind" and "Happy" using a well-vetted psychometric instrument - the GPOMS (Google profit of mood state) algorithm. They trained a SOFNN (Self-Organizing Fuzzy Neural Network) and showed that one of the six mood dimensions called "Calm" was a statistically significant mood predictor for the DJIA daily price up and down change. However, (Bollen et al., 2011b)'s research only predicted movements in the DJIA index but it was not clear in which individual companies the user should invest.

3 Method

In this paper, we illustrate a hybrid method to train a series of classifiers for predicting the polarity of the daily market opening price change for four tech stocks namely Apple (AAPL), Google (GOOG), Microsoft (MSFT) and Amazon (AMZN). These were chosen because related topics such as their products, famous staff, etc. are all actively discussed in the social media. We also looked at other tech companies like Research In Motion Limited (RIMM), Yahoo (YHOO), etc. but found that the number of Tweets related to these companies is relatively small. Because we downloaded daily Tweets using the Twitter online streaming API called Firehose[1], we only had access to 1% of the Twitter corpus. Therefore, we expect the technique presented here to apply on a larger scale when we are able to access more of the Twitter corpus. Figure 1 shows an overview of the stock market prediction model.

[1] https://dev.twitter.com/docs/streaming-apis/streams/public

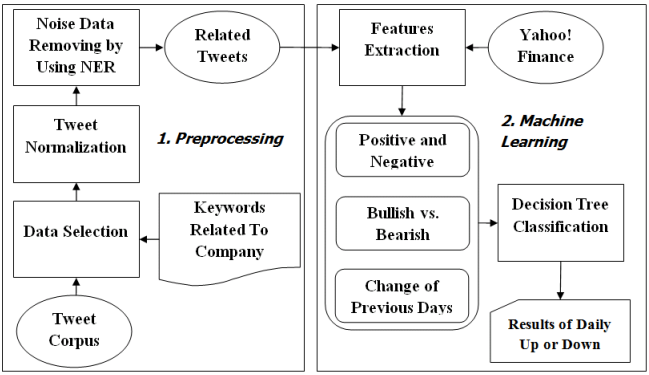

Figure 1: Daily up and down stock market prediction model

3.1 Data

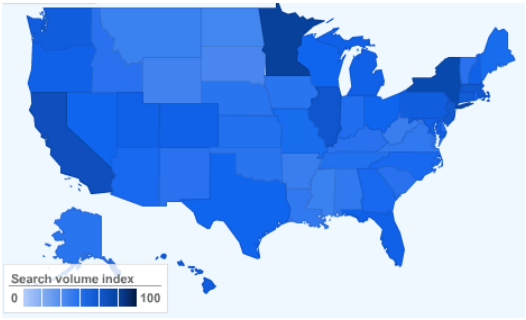

Figure 2: Keyword searching heat map (Apple).

Data containing 5,001,460 daily Tweets was crawled by using Twitter online streaming API from 1^{st} April 2011 to 31^{st} May 2011. In this initial investigation we would like to control the market sample to focus on the United States, so we geographically focused our Twitter queries on four large cities: New York, Chicago, Los Angeles and San Francisco. This also has some benefit in harmonising vocabulary – a recent study by (Gouws et al., 2011) noted significant differences in word abbreviation behaviour between British and American microtext authors. The Google Insights heat map[2] shown in figure 2 indicates that the people living in these four areas are more interested in the four companies. To compute the daily up and down of stock

[2]http://www.google.com/insights/search

market changes we employed Yahoo! Finance data using (1).

$$Move_i = OpenP_i - CloseP_{Pre(i)} = \begin{cases} \text{difference} >= 0 & \text{Raise} \\ \text{difference} < 0 & \text{Drop} \end{cases} \quad (1)$$

where i is a stock market open day, $Move_i$ is stock market change on day i, $OpenP_i$ is the opening price on day i, $CloseP_j$ is the closing price of a stock on day j and $Pre(i)$ is previous stock market open day of day i. In total we have 41 days of stock market open days of data excluding weekends and public holidays.

3.2 Time normalization

Before processing, we normalised the original Tweet time stamps to American Eastern Time, i.e. the NASDAQ stock market time, also taking into account changes in Winter time and Summer time. This data was then used to predict the following day's up or down movement in the four target stocks.

3.3 Preprocessing

3.3.1 Data selection

In this step, we aim to filter out those Tweets that are not relevant to the four target stocks. For the sake of simplicity this step introduces an assumption that the market as a whole (e.g. of mobile telephones) does not influence the stock price of an individual company. Starting from the names of the famous products of the four companies, we performed keyword more than factors we explore here expansion using Google Adwords[3] to create a list of terms for each stock. We filtered again to include only Tweets where users mention one of keywords. For example:

- AAPL: iphone, ipad, mac, ipod, ios, siri, apple store, itunes, etc.
- GOOG: google, gmail, gtalk, chrome, android, youtube, etc.
- MSFT: windows 7, windows 8, xbox, internet explorer, visual studio, office 2007, office 2010, etc.
- AMZN: alexa internet, amazon.com, amazon kindle, imdb, lovefilm, etc.

After keyword selection we can query for Tweets related to the companies.

3.3.2 Tweet normalization

When dealing with Tweets as has been reported in (Han and Baldwin, 2011), there is an issue of lexical normalization for out-of-vocabulary (OOV) words. Because around 15% of Tweets have 50% or more OOV tokens (Han and Baldwin, 2011), we applied the method proposed by (Han et al., 2012) to normalize Tweets. For example, with an input *"Ipad 2 is very coooooooooooooooollllllllllllllllll"*

[3] https://adwords.google.com

- **First step:** each multiple character will be reduced to three. With the example, the output of this step is *"Ipad 2 is very cooolll"*
- **Second step:** apply normalization lexicon proposed by (Han et al., 2012) to normalize Tweets. For the example, the output of this step is *"Ipad 2 is very cool"*

We also normalized Tweet meta data, that is, every link becomes *LINK* and every account name becomes *ACCOUNT*. Hash tags are treated as normal words.

3.3.3 Noise data removing

After the normalization step, we wanted to identify Tweets on the topic of tech products, For example, although *"LOL YES !! RT : *ACCOUNT* : *ACCOUNT* You know we loved your Mac & Cheese Tweet. Can we turn it into a National television ad right now ?."* contains the *"mac"* keyword, it isn't a product of Apple corporation.

To resolve this problem, we built a Named Entity Recognition(NER) system to identify whether the Tweet contains name entities related to the companies or not based on a linear Conditional Random Fields(CRF) model. The Linear CRF model is used because it is well-studied and has been successfully used in state-of-the-art NER systems (Finkel et al., 2005)(Finkel and Manning, 2009)(Wang, 2009). If the Tweet doesn't contain any named entities as listed on the company keyword list, it is removed.

Twitter users are interested in named entities, such as, famous entrepreneurs, organization names, trendy hardware and software when they talk about tech companies. We collected and labelled manually 3665 randomly sampled Tweets related to the companies based on keywords. These included 280 people names (42 unique), 395 organization names (38 unique), 2528 hardware names (171 unique) and 1401 software names (294 unique). Overall, we have 4604 named entities in which 540 entities are unique.

Named entity recognition task

Given a Tweet as input, our task is to identify both the boundary and the class of each mention of entities of predefined types. We focus on four types of entities in our study, namely, persons, organizations, hardware, and software.

The following example illustrates our task. The input is "Only Apple can be surprising at not being surprising. I must be immune to the reality distortion field. Tell me when Lion & iOS 5 are out" The expected output is as follows:"Only <Organization>Apple</Organization> can be surprising at not being surprising. I must be immune to the reality distortion field. Tell me when <Software>Lion</Software> &; <Software>iOS 5</Software> are out", meaning that "Apple" is an organization, while "Lion" and "iOS 5" are software.

In our experiments, the CRF++[4] toolkit is used to train a linear CRF model. For each word, our CRF model extracts orthographic and lexical features based on (Wang, 2009) as follows:

- **Orthographic Features:** Word forms were mapped to a small number of standard orthographic classes. The present model uses 8 orthographic features to indicate whether the words are capitalised or upper case, whether they are alphanumeric or contain any slashes, whether the words are number or date, and whether the words are emails or punctuation marks.

[4] http://crfpp.sourceforge.net/

- **Lexical Features:** Every token in the training data was used as a feature. Alphabetic words in the training data were converted to lowercase, spelling errors detected in the proofreading stage were replaced by the correct resolution using the same technique in Tweet normalization step. Shorthand and abbreviations were expanded into bag of words (BOW) features. To utilise the context information, neighbouring words in the window [-2, +2] are also added as features $w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}$ where w_i is target word. A context window size of 2 is chosen because it yields the best performance.

After removing noise data, finally, the corpus for each company contained the following numbers of Tweets: AAPL (18,317), GOOG (28,435), AMZN (35,324), MSFT (4,023).

3.4 Machine Learning Framework

Following the main idea of behavioral finance theory (BFT) (Smith, 2003) and the efficient market hypothesis (EMH) (Fama, 1965), we make the assumption that the stock price can be predicted by using some features namely (1) sentiment related to the company: positive or negative, (2) the degree of market confidence: bullish or bearish. We then trained a Decision Tree(C4.5) classifier (Quinlan, 1993) by combining these features to address the text binary classification problem for predicting the daily up and down changes using our stock message set. Decision trees(Quinlan, 1993) have been widely used for prediction problems, often with good results (Cheung Chiu and Webb, 1998)(Wang and Chan, 2006).

Positive	:), :-), xd, (:, :p, :-p, ;), ;-), etc.
Negative	):, :(, :[, ;(, :{, ;'(, :'(, etc.

Table 1: Lexicon of emoticons

Positive	**Examples**
APPL	1. God I love my Iphone
	2. Yahoo!!! I finally bought the MacBook pro!!! :)
GOOG	3. Man, Google maps transit is awesome.
	4. Loving gmail motion and helvetica :)
MSFT	5. ...the Xbox & the Xbox 360 are the best designed consoles ever...
	6. hmm. i like this internet explorer 9
AMZN	7. Just saw this on Amazon: 'Kindle Wi-Fi' ... Cool:)
	8. got three books in the mail today. thank you amazon...
Negative	**Examples**
APPL	1. iPod battery low :(
	2. My iPhone no longer works on 3G networks... Boo :(
GOOG	3. my google chrome is being weird right now...:(
	4. New Google maps is very confused...
MSFT	5. Two things to never trust in Microsoft Word : spell check and grammar
	6. God hates me, and I hate Microsoft
AMZN	7. ...new sites that won't load due to the big Amazon AWS failure. :(
	8. Amazon servers went down, taking HootSuite and Foursquare down with it. :(

Table 2: Examples of positive/negative Tweets

3.4.1 Positive and Negative(Pos_Neg) features

To detect the sentiment of Tweets posted by Twitter users, we employ an online sentiment classifier called Twitter Sentiment Tool (TST)(Go et al., 2009). This tool is able to determine the polarity for a certain Tweet using a keyword-based algorithm. The polarity set contains three elements positive, neutral, negative. It is helpful to maximize the polarity distance between Tweets since we only need to be sure a query is positive or negative, and we ignore other Tweets (neutral). Furthermore, this online tool is ideally suited to our task since it is trained specifically on Twitter data. However, we still make a simple improvement before we send Tweets to TST. For example: *Caught up on the dvr. Time for bed. Must do laundry tomorrow, before I go and get my new ipad.:)*, this is obviously a positive sentiment for Apple but it is interesting that TST classifies this Tweet as Neutral. The designers (Go et al., 2009) of TST use emoticons as noisy labels to train the sentiment classifier so we expect this example to be Positive due to the ":)". Following from (Go et al., 2009)'s idea we created an emoticon lexicon shown in Table 1 to identify positive or negative of Tweets before we send Tweets to TST. Table 2 shows examples of positive/negative Tweets. After checking against the lexicon of emoticons in Table 1 and classifying with TST, we simply aggregate the number of positives and negatives for each day:

Positive feature is identified using (2,3).

$$PosDiff_i = \sum_i positive_i - \sum_{i-1} positive_{i-1} \tag{2}$$

$$Positive(D)_i = \begin{cases} 1 & \text{If } PosDiff_i >= 0 \\ 0 & \text{If } PosDiff_i < 0 \end{cases} \tag{3}$$

Where $positive_i$ denotes the number of positively classified message by TST for a particular company on day i. Similarly, the negative feature is identified by functions 4, 5.

$$NegDiff_i = \sum_i negative_i - \sum_{i-1} negative_{i-1} \tag{4}$$

$$Negative(D)_i = \begin{cases} 1 & \text{If } NegDiff_i >= 0 \\ 0 & \text{If } NegDiff_i < 0 \end{cases} \tag{5}$$

Where $negative_i$ denotes the number of negatively classified message by TST for a particular company on day i.

3.4.2 Bullish vs. Bearish features

To determine whether consumers have market confidence in the company, we made use of a Part-of-speech (POS) tagger to extract adjective, noun, adverb and verb words and fixed them to "bullish" and "bearish" as anchor words. We chose CMU POS Tagger proposed by (Gimpel et al., 2011) because this POS Tagger achieved the state of the art on Tweet data. We then calculated the anchor words using the Semantic Orientation (SO) algorithm (Turney, 2002). This algorithm uses mutual information to measure the association between two words. The

Pointwise Mutual Information (PMI) is formulated by function 6.

$$PMI(w_1, w_2) = \log_2(\frac{p(w_1 \& w_2)}{p(w_1)p(w_2)}) \tag{6}$$

where w_1 and w_2 are two word strings. $p(w_1 \& w_2)$ are the probability that both words co-occurred. $p(w_1)$ and $p(w_2)$ are the probability that an isolated word occurs. The ratio shows a metric of statistical dependence between w_1 and w_2. Thus, SO can be defined by (7).

$$SO(w) = PMI(w, \text{“}bullish'') - PMI(w, \text{“}bearish'') \tag{7}$$

Here, based on our anchor words, we redefined SO in (8).

$$SO(w) = \log_2(\frac{\#(wNEAR\text{“}bullish'')\#(\text{“}bearish'')}{\#(wNEAR\text{“}bearish'')\#(\text{“}bullish'')}) \tag{8}$$

To compute the equation 8, we used the AltaVista[5] search engine for the following reasons: (1) AltaVista only contains English webpages; (2) There is a location restriction that can allow us to focus on United States webpages, and (3) It provides a “NEAR” operator that helps to find documents containing pairs of words within 10 words distance of each other. (Turney, 2002) noted that the performance of the NEAR operator was better than the AND operator as a semantic association measure.

To avoid division by zero, we applied an add-one data smoothing method in queries whose result from AltaVista is zero, that is, a null hit. After this online training process, we build up a dictionary of four different POST: adjectives, nouns, adverbs, and verbs.

The bullish-bearish feature of Tweet day i is identified by functions 9, 10.

$$Bullish\_Bearish\_Diff_i = Bullish\_bearish_i - Bullish\_bearish_{i-1} \tag{9}$$

$$Bullish\_Bearish(D)_i = \begin{cases} 1 & \text{If } Bullish\_Bearish\_Diff_i >= 0 \\ 0 & \text{If } Bullish\_Bearish\_Diff_i < 0 \end{cases} \tag{10}$$

Where $Bullish\_Bearish_i$ denotes the mean of SO values of all extracted words in day i.

3.4.3 Stock market changes on price previous days

To predict stock market movement on day i, we applied in previous days $(i-1, i-2, i-3, ...)$as features to train the classifier. Number of previous days is experimentally identified as 3 to yield the highest performance.

Finally, the changes of three previous days were combined with the previous fixed features - positive, negative and bullish/bearish to get the best performance calculated by the Decision Tree classifier(Quinlan, 1993).

[5] http://www.altavista.com/

4 Experimental Results

4.1 Main results

Due to the limited corpus size of 41 days in the training set, and also considering the over-fitting problem, we chose the 10-fold cross validation method to estimate the accuracy of our approach. We trained our classifier using Decision Tree (C4.5) with features generated from previous day data. This is because our analysis shown in Table 3 indicates that Pos_Neg features have the lowest p-value in the Granger-Causality Test with 1 day lag. Granger causality analysis rests on the assumption that if a variable X causes Y then changes in X will systematically occur before changes in Y . We will thus find that the lagged values of X will exhibit a statistically significant correlation with Y . Correlation however does not prove causation. We therefore use Granger causality analysis in a similar fashion to (Gilbert and Karahalios, 2010); we are not testing actual causation but whether one time series has predictive information about the other or not. All methods used filtering with NER. Table 4 shows daily prediction accuracy on Apple Inc., Google, Amazon, and Microsoft stock prices respectively.

Lag	AAPL		GOOG		MSFT		AMZN	
	PF	**NF**	**PF**	**NF**	**PF**	**NF**	**PF**	**NF**
1 day	**0.004****	0.144	0.107	**0.018***	0.539	**0.34**	**0.032***	0.76
2 days	**0.012***	0.24	0.4	0.12	0.96	0.81	0.11	0.83
3 days	0.051	0.44	0.265	0.072	0.865	0.879	0.074	0.749
4 days	0.056	0.589	0.587	0.289	0.924	0.984	0.156	0.829
5 days	0.102	0.326	0.241	0.143	0.975	0.713	0.443	0.877
6 days	0.1	0.361	0.095	0.156	0.981	0.775	0.282	0.576
7 days	0.27	0.47	0.119	0.3	0.903	0.781	0.406	0.63

Table 3: Statistical significance (P-Values) of Granger-Causality Test between Pos_Neg features and stock market movement of four companies in period April 1, 2011 to May 31, 2011. ***PF:*** **Positive Feature,** ***NF:*** **Negative Feature (p-value < 0.01: **, p-value < 0.05:** *)

Method	AAPL	GOOG	MSFT	AMZN
Only bullish/bearish	53.66%	58.54%	56.10%	37.50%
Only previous days	51.22%	53.66%	73.73%	37.50%
Only Pos_Neg	73.17%	68.29%	56.10%	71.88%
Bullish/bearish + previous days	63.41%	63.41%	**75.61**%	62.50%
Bullish/bearish + Pos_Neg	73.17%	70.73%	56.10%	71.88%
Pos_Neg + previous days	73.17%	68.29%	70.73%	71.88%
Pos_Neg + bullish/bearish + previous days	**82.93**%	**80.49**%	**75.61**%	**75.00%**

Table 4: Prediction accuracy on each stock

The results shown in Table 4 indicate a surprisingly high level of accuracy for stock price polarity prediction using our proposed model. The combination of all Pos_Neg, bullish/bearish, and previous change yields superior performance for all stocks.

No single feature obviously stands out as superior in all Apple, Google, Microsoft and Amazon stocks. Pos_Neg features can predict well for Apple, Google, Amazon with accuracies of 73.17%, 68.29%, and 71.88% but result is a fall for Microsoft with accuracy of only 56.10%. The accuracies consistent to Granger Causality Test's results shown in Table 3 that we can reject

NULL hypothesis that the mood time series does not predict APPL, GOOG, and AMZN stock markets (P-value < 0.05). Pos_Neg features cannot predict well Microsoft stock because the number of Pos_Neg Tweets is very few (over 20 days have no Pos_Neg Tweets). So in the case of frequent Pos_Neg Tweets, Pos_Neg features appear to function as a strong prediction of stock market movement.

The previous day's price movement and Bullish/bearish features seem to offer explicitly weaker predictability. However when we combine these two features, the predictability of our system improves significantly, for example, from 37.50% to 62.50% on Amazon stock. The other combination of two features can slightly increase our system's predictability, for example, the combination of Pos_Neg and Bullish/bearish features made an improvement of Google's accuracy from 68.29% to 70.73%.

The combination of Previous days's price movement, Bullish/bearish and Pos_Neg features create a superior model in all Apple, Google, Microsoft and Amazon stocks. Accuracies for our stock market prediction system increase to a peak of 82.93%, 80.49%, 75.61% and 75.00% in Apple, Google, Microsoft and Amazon respectively.

To show the effectiveness and correctness of the proposed model, we applied the model with a combination of Previous days's price movement, Bullish/bearish and Pos_Neg features to an online test in which we accessed realtime Tweets using the Twitter online streaming API. The online test was implemented from 8^{th} September 2012 to 26^{th} September 2012. Table 5 shows the experimental results of the online test with high prediction accuracies of 76.92%, 76.92%, 69.23% and 84.62% in Apple, Google, Microsoft and Amazon respectively. The online experimental result provides an additional indication of the effectiveness and correctness of our proposed model.

Stock market	**Accuracy**
AAPL	76.92%
GOOG	76.92%
MSFT	69.23%
AMZN	84.62%

Table 5: Experimental results in the online test

We note that the results offer a company analysis level in contrast with (Bollen et al., 2011b)'s research. Although (Bollen et al., 2011b) achieved accuracy of 87.6% in predicting the daily up and down changes in the closing values of the Dow Jones industrial average, it was not clear which companies the user should invest in. Again, in contrast to (Bollen et al., 2011b), we do not need specialized sentiment lexicons. Although the result is not directly comparable with (Bar-Haim et al., 2011), because of differences in the size of the data set and the number of stocks studied, it provides *prima facie* evidence for a much higher level of predictive performance using sentiment related to company specific features over identification of expert stock pickers. The result also indicates that company related sentiment can offer strong correlation to individual stock prices in contrast to (Das and Chen, 2007)'s experience with stock board messages. The number of days and the number of stocks in our experiment should make us cautious about making strong conclusion from the results. Nevertheless we believe the result are indicative of interesting trends that should be followed up in future works.

Entity Type	Precision	Recall	F-score
Hardware	97.06%	84.33%	90.24%
Software	94.78%	70.78%	81.02%
Organization	92.82%	66.51%	77.03%
Person	100%	81.12%	89.20%
All Entities	**90.02%**	**78.08%**	**83.60%**

Table 6: Overall NER experimental results

Stock market	Accuracy with NER	Accuracy without NER
AAPL	**82.93%**	73.17%
GOOG	**80.49%**	75.61%
MSFT	**75.61%**	68.29%
AMZN	**75.00%**	71.88%

Table 7: Effectiveness of Using Named Entity Recognition in tech stock market prediction by all using Pos_Neg, bullish/bearish and previous days features

4.2 NER experimental results

We randomly selected 3665 Tweets related to the companies using keywords. After that, the Tweets were labeled manually by one of the authors, so that the beginning and the end of each named entity is marked as <TYPE> and </TYPE>, respectively. This then formed the gold-standard data set. Here TYPE is SOFTWARE, HARDWARE, ORGANIZATION, or PERSON. The gold-standard data set is evenly split into ten parts to implement ten folds test: nine for training and one for testing. We use Precision, Recall, and F-score as the evaluation metrics. Precision is a measure of the percentage of correct output labels, and recall tells us the percentage correctly labelled in the gold-standard data set, while F1 is the harmonic mean of precision and recall. The NER experimental results are showed in Table 6.

The overall NER experimental result is good with Precision of 90.02%, Recall of 78.08%, and F-score of 83.60%. We can achieve high performance in the NER system for the following reasons. Firstly, although 3665 Tweets were used as the golden corpus to train and test the NER system, there are 280 people names (42 unique), 395 organization names (38 unique), 2528 hardware names (171 unique) and 1401 software names (294 unique). Overall, we have 4604 named entities in which 540 entities are unique. So it made the training set cover most contexts of these entities. Secondly, the Tweet's informal nature causes the low performance of recent NER systems on Tweet data (Liu et al., 2011). To overtake that problem, we use normalization technique (Han and Baldwin, 2011) on Tweet data before applying the NER system, leading to the high performance.

4.3 Effects of NER to remove noise

Table 7 shows the performance of the best performing method in Table 4 (Pos_Neg + bullish/bearish + previous days) with and without using NER system to remove noise. NER filter step plays a very important role in our system. Without this step, the performance of our system decreases significantly to 73.17%, 75.61%, 68.29%, and 71.88% for all stocks. We further check the data before and after the noise-removal step. There were many unrelated Pos_Neg Tweets removed for all companies. It helps the identification of positive and negative features by functions 2,3,4,5 and bullish vs bearish feature by functions 6,7,8 to be more

accurate.

5 Conclusion and Future Work

In this paper we have addressed the problem of predicting daily up and down movements in individual tech stock prices using a combination of three feature types from Twitter messages related to the company: positive and negative sentiment, consumer confidence in the product with respect to bullish and bearish lexicon, and the change on three previous days of stock market price. The features were employed in a Decision Tree(C4.5) classifier to yield high levels of accuracies of 82.93%,80.49%, 75.61% and 75.00% in predicting the daily up and down changes of Apple (AAPL), Google (GOOG), Microsoft (MSFT) and Amazon (AMZN) stocks respectively. We also indicated the influence of each features to the results of our proposed model. Especially, the experimental results showed that using NER to remove noise data played a very important role in the stock market prediction model.

The study we have presented has been limited to 41 days of Tweets so we must regard any results as indicative rather than conclusive. In future work, with access to more data, we would like to expand our investigation to a longer time frame and a wider range of companies as well as looking at shorter market durations.

Acknowledgments

This work was done when the first author was an internship student in the National Institute of Informatics(NII), Tokyo, Japan. And this work was supported by NII. We thank the reviewers for their valuable comments.

References

Bar-Haim, R., Dinur, E., Feldman, R., Fresko, M., and Goldstein, G. (2011). Identifying and following expert investors in stock microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1310–1319, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bollen, J., Gonçalves, B., Ruan, G., and Mao, H. (2011a). Happiness is assortative in online social networks. *CoRR*, abs/1103.0784.

Bollen, J., Mao, H., and Zeng, X. (2011b). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8.

Cheung Chiu, B. and Webb, G. I. (1998). Using decision trees for agent modeling: Improving prediction performance. *User Modeling and User-Adapted Interaction*, 8(1-2):131–152.

Das, S. R. and Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the Web. volume 53, pages 1375–1388.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA. ACM.

Fama, E. F. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, 38(1):34–105.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting*

on Association for Computational Linguistics, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Finkel, J. R. and Manning, C. D. (2009). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 141–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gilbert, E. and Karahalios, K. (2010). Widespread worry and the stock market. In *In Proceedings of the International Conference on Weblogs and Social*.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. Technical report, Stanford University.

Gouws, S., Metzler, D., Cai, C., and Hovy, E. (2011). Contextual bearing on linguistic variation in social media. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 20–29, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han, B., Cook, P., and Baldwin, T. (2012). Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 421–432, Jeju Island, Korea.

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics - Volume 1*, COLING '00, pages 299–305, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA. ACM.

Kivran-Swaine, F. and Naaman, M. (2011). Network properties and social sharing of emotions in social awareness streams. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, pages 379–382, New York, NY, USA. ACM.

Krishnamurthy, B., Gill, P., and Arlitt, M. (2008). A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 19–24, New York, NY, USA. ACM.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA. ACM.

Lin, C. X., Zhao, B., Mei, Q., and Han, J. (2010). Pet: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 929–938, New York, NY, USA. ACM.

Liu, X., Zhang, S., Wei, F., and Zhou, M. (2011). Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 359–367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 412–418.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 181–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Petrovic, S., Osborne, M., and Lavrenko, V. (2012). Using paraphrases for improving first story detection in news and twitter. In *HLT-NAACL*, pages 338–346.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Ritterman, J., Osborne, M., and Klein, E. (2009). Using prediction markets and Twitter to predict a swine flu pandemic. In *Proceedings of the 1st International Workshop on Mining Social Media*.

Smith, V. L. (2003). Constructivist and ecological rationality in economics. pages 502–561.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, J.-L. and Chan, S.-H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Syst. Appl.*, 30(4):605–611.

Wang, Y. (2009). Annotating and recognising named entities in clinical notes. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 18–26, Suntec, Singapore. Association for Computational Linguistics.

Wiebe, J., Bruce, R., Bell, M., Martin, M., and Wilson, T. (2001). A corpus study of evaluative and speculative language. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhao, Q., Mitra, P., and Chen, B. (2007). Temporal and information flow based event detection from social text streams. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, AAAI'07, pages 1501–1506. AAAI Press.