IJCSN International Journal of Computer Science and Network, Volume 4, Issue 2, April 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

Prediction of Stock Market Shift using Sentiment Analysis of Twitter Feeds, Clustering and Ranking

[1] Tejas Sathe, [2] Siddhartha Gupta, [3] Shreya Nair, [4] Sukhada Bhingarkar

[1,2,3,4] Dept. of Computer Engineering
MIT College of Engineering
Paud Road, Pune.

Abstract - The stock market is fluctuating constantly. The rise and fall in stock prices are seemingly random. However, this is not so. Even a minute happening in the company can have a huge effect on the stock price. As each investor buys and sells the stock, the price rises and falls depending on the sale and purchase, the demand and supply. Whether or not an investor buys a particular company's stock is based on his knowledge and impression of the company. The latter is what we will employ to decide whether or not to buy a certain company's stock at the current price. There are 6 accepted discrete moods. Millions of people tweet every second. A fairly accurate prediction and analysis of the tweet's underlying mood can be made using sentiment analysis. Each word has a certain grammatical signature that tells us which mood it belongs to. Depending on what the users are feeling about a company as they tweet about it this engine will decide whether or not one should buy stocks of that company.
This paper describes how to map this mood with market sentiment and in turn with prediction of rise/fall of stock prices.

Keywords – *Twitter, Sentiment Analysis, Successive Deviations, Stock Market, Mood.*

1. Introduction

Sentiment analysis aims to determine the attitude of a speaker or a writer (in this case the person who is tweeting) with respect to some topic (in this case the stock market). Sentiment Analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials.[1] Stock market being a volatile market, it isn't possible to predict the stock market accurately with the help of standard and existing algorithms. As the mood of the market is set by people's emotions towards the market, analyzing the tweets would give us an overview of the people's emotions which would help us predict if the market is in bullish or bearish mood. A bull market is when the market is showing confidence with the prices going up. A bear market is the opposite of bull market. A bear market shows low confidence and in this market the prices drop.[2] Stock market plays a crucial role in the economy of a country. They are an integral part of all major economies. They provide unique services and benefits to corporations, individual investors and governments.[3] Billions of dollars are traded on different stock exchanges everyday.[4] Being able to get a foresight of the direction the stock market is moving will be financially beneficial. We will do this for a large number of people. Depending on the overall nature of sentiment (positive or negative) we will attempt to predict whether a particular stock will rise or fall. We will test this against a mock portfolio to establish the validity of the hypothesis.

The remaining paper is organized as follows:
Section 2 describes our research into all the related work that has already been done in this field. Section 3 gives a brief overview of the architecture of our engine. Section 4 gives a detailed step by step explanation of our implementation. Section 5 underlines the testing strategy we undertook and the results it produced. Section 6 concludes the paper and presents the future work that can be done.

2. Related Work

Sentiment Analysis as an area is on the rise. It comes under the Natural Language Processing field. It ranges from document level classification to learning the polarity (positive or negative connotation) of words and phrases. Turney and Pang performed some of the earliest work in this field. They applied different methods for detecting the polarity of product reviews and movie reviews respectively. This was at the document level. A document's polarity can also be classified, which was done by Pang and Snyder. Pang expanded the basic task of classifying a movie review as either positive or negative to predicting star ratings on either a 3 or a 4 star scale, while Snyder performed an in-depth analysis of restaurant reviews, predicting ratings for various aspects of the given restaurant, such

as the food and atmosphere (on a five-star scale). A different method for determining sentiment is the use of a scaling system whereby words commonly associated with having a negative, neutral or positive sentiment with them are given an associated number on a -10 to +10 scale (most negative up to most positive) and when a piece of unstructured text is analyzed using Natural Language Processing. Depending on the score, the polarity of that word is determined.

Another research direction is the analysis of *subjectivity* and *objectivity*. This involves the classification of a piece of text as either subjective or objective. This is generally more difficult than polarity determination. The subjectivity of words and phrases may depend on their context and an objective document may contain subjective sentences (e.g., a news article quoting people's opinions).

Yet another analysis model is the *feature/aspect-based sentiment analysis* model. It refers to determining the opinions or sentiments expressed on different features or aspects of entities, e.g., of a cell phone, a digital camera, or a bank. A feature or aspect is an attribute or component of an entity, e.g., the screen of a cell phone, or the picture quality of a camera. This problem involves several sub-problems, e.g., identifying relevant entities, extracting their features/aspects, and determining whether an opinion expressed on each feature/aspect is positive, negative or neutral.

Modern sentiment analysis can be used for a variety of things. Open source software tools deploy machine learning, statistics, and natural language processing techniques to automate sentiment analysis on large collections of texts, including web pages, online news, internet discussion groups, online reviews, web blogs, and social media. As automated systems cannot account for historical tendencies, a human component to sentiment analysis is essential. Automation can only accurately predict 23% of comments correctly classified by humans. Sometimes, the structure of comments and sentiments is fairly complex. Also, the problem of sentiment analysis is non-monotonic in respect to sentence extension and stop-word substitution (compare *THEY would not let my dog stay in this hotel* vs *I would not let my dog stay in this hotel*). To address this issue a number of rule-based and reasoning-based approaches have been applied to sentiment analysis, including Defeasible Logic Programming.

Thus, all in all, modern sentiment analysis can predict the mood of a statement fairly accurately. There are 6 distinct, discernible moods that a statement can be classified into.

As Twitter imposes a restriction on the number of characters per tweet, this process becomes somewhat simpler as the size of the data is reduced. Various ways of analyzing twitter sentiments have cropped up. Several researchers and companies rely on the analysis of emoticons for prediction mood of the sentence[9]. This mood or connotation has been used in various ways to predict future trends. Godbole et al [8] propose several parameters (subjective as well as objective) to achieve this end.

They define the following terms:

Polarity: Tells us whether the mood/connotation of the sentence is positive or negative.

Subjectivity: Tells us exactly how much sentiment is contained in a particular sentence.

world_polarity = (positive sentiment references) / (total sentiment references)

entity_polarity, however, is calculated only for a particular day (day_i).

$entity\_polarity_i$ = *(positive sentiment references*$_i$*) / (total sentiment references*$_i$*)*

These two parameters will accurately tell us the overall mood of a time period. [8]

Furthermore, moods can also be guessed using emoticons, which are cartoon faces with human expressions. They are classified into happy and sad emoticons. These are hard coded, and the underlying symbols have no significance in determining the mood depicted by the emoticon.

Happy emoticons: ":-)", ":)", "=)", ":D" etc.
Sad emoticons: ":-(", ":(", "=(", ";(" etc.

These will tell a classifier to understand a sentence's mood as positive or negative based on the accompanying emoticon [7]. This is extremely accurate, as no sane person would post a happy emoticon with a sad sentence or vice versa, or post both types of emoticons in the same sentence.

Twitter is one of the biggest social networks that exists today. People like to post minutiae of their lives on such networks. While some people might argue that this is irritating and a waste of bandwidth, time and who knows what else, many entities certainly use it to their advantage. For example, such details of people's lives gives a rare insight into what they like and dislike, what they subscribe to, where they shop and shop and so much more. This data is analyzed and used by companies to advertize their product to only a select number of people. This gives them a target audience, increasing the odds of their product being sold. This can be easily achieved by Twitter's public domain hashtag

data set. People post relevant hashtags at the end of each tweet. These can be used to gauge their interests. This dataset is a subset of the Edinburg Corpus dataset[9]. This can also be used to predict stock market shift, feeding upon people's satisfaction or discontentedness with a particular company. Chen et al [5] propose using this to determine not only the direction but also the intensity of the shift. Based on this they come up with a suitable investment strategy that they suggest to the user, one in which they feel the user will have maximum monetary gain.

3. System Architecture

This diagram gives a brief overview of the architecture of our engine.

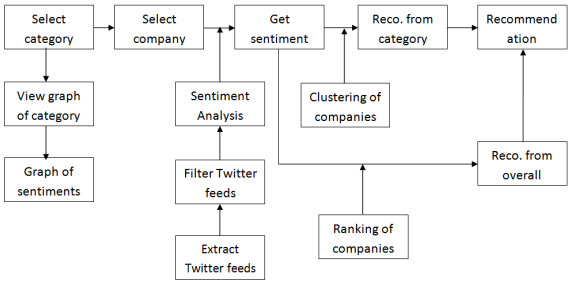

Fig 1: System Architecture

First, the user selects a category (e.g. Banks). Then he can view a graph of sentiments of all banks on the stock market. Alternatively, he can select a company belonging to that category (e.g. ICICI Bank). He gets a recommendation on whether or not to buy stock based on analysis of market sentiment about that company. This is done by extracting tweets pertaining to that company, filtering them to remove things that don't affect sentiment (i.e. hyperlinks, hashtags, handles and numbers) and perform sentiment analysis on them.

He gets a recommendation as to what percent of his budget he should invest in that company. Alternatively, he also gets two choices: Category Recommendation (similar companies from the same category) and Overall Recommendation (similar companies from the overall list). Recommendations from the same category are generated using clustering (explained later) and overall recommendations are generated using ranking (explained later).

4. Implementation

The flow of the implementation is as follows: Extraction of tweets (getting tweets pertaining to particular company), Sentiment Analysis (analyzing set of tweets for overall sentiment), Investment Strategy (coming up with an investment scheme based on the results of the aforementioned sentiment analysis), Clustering (to generate recommendations from the same category) and Ranking (to generate recommendations from the overall list of companies.

4.1. Extraction of Tweets

This is an integral part of the process. As Sentiment Analysis is performed on tweets, this is the most important part.

We're using Python's Twitter API for extraction. After registering your web application with Twitter, you receive two keys: customer key and API key. Using OAuth (which is used to log into sites using other sites, e.g. login using Facebook), the web app is recognized based on the two keys and the tweet database can be queried. The API we are using is called tweepy. It allows us to access tweets from up to a week ago. We extract tweets and perform Sentiment Analysis on them.

4.2. Sentiment Analysis

SentiWordNet has generated a text file that contains positive and negative scores of almost all words with each of its connotations. We extracted and stored the data in this file into a database to simplify the process. To decide which connotation best fits the sentence, we use successive deviation.

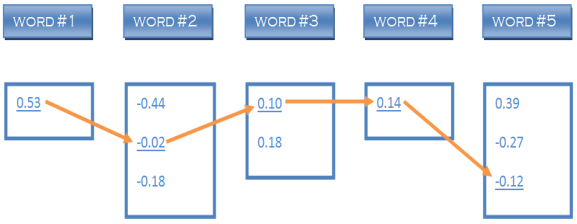

Fig 2: Successive deviation

This process decides which connotation will be best in the context of the sentence. For the first two words, each score of the first word is compared with each score of the second word. The pair with the least deviation is selected. From the third word onwards, the score with the least deviation from the earlier one is selected. Then these scores are simply added to get the sentiment of the sentence, or in this case, the tweet. This same process is run on the scores of tweets that we scan and get the total sentiment of the set of tweets.

4.3. Investment Strategy

We came up with a random investment strategy, i.e. **k*s%** where **k** is a constant and **s** is the difference between the current sentiment and the most recent sentiment of that company (the one obtained when someone last checked that company on our engine). We

wrote an automated script that recalculates **k** thrice a day based on the following rules:

Initially, **k** is set to 1. If **s** is positive/negative it signifies that we predict the company stock price will rise/fall respectively. If our prediction is correct, **k** is left unchanged.

If **s** is positive and the price falls, we *decrement* **k** by 0.1. If **s** is negative and the price rises, we *increment* **k** by 0.1. This calculation is done thrice a day.

4.4. Clustering

This technique is used to generate recommendations from the same category. Here we use an adapted version of the k-means clustering algorithm. In the k-means, we begin with random means and put elements in clusters based on which mean they're closer to. Once we do, we recalculate the means and continue this process. We stop when the old means and the corresponding new means are equal.

However, in our method, when we recalculate the means, instead of using those as the new means, we check the clusters for elements that are closest to them and use those as the new means. Thus, in each iteration, we select a company as a mean and compare others using that as reference.

Clustering is done separately based on two parameters: sentiment and count. Count is the number of times a company has been checked on our engine. Each time someone requests a recommendation for that company, count is incremented by 1.

Thus, we end up with 6 clusters:
sentiment: top cluster, mean companies, bottom cluster
count: top cluster, mean companies, bottom cluster

From these we generate three final clusters: **high** highly recommended), **mid** (moderately recommended) and **low** (not recommended). We determine these by taking intersections of various combinations of the above 6 clusters. e.g. If a company is in the top cluster based on sentiment and in the bottom based on count, we place it in *mid*. The list of such rules:

top + top = high
top + bottom = bottom + top = mid
bottom + bottom = low
mean + top = top + mean = high
mean + low = low + mean = mid
mean + mean = mid

4.5. Ranking

We use ranking to generate recommendation from the overall list of companies, not restricted to the same category.

We store the companies each user has looked at and generate a mapping. e.g.

U1: {C1, C6, C2, C7, C3}
U2: {C2, C3, C1, C5, C9}

Here **U** stands for User and **C** stands for Company. We find commonly occurring patterns and measure their frequency. e.g.

{C1, C2}: 4
{C1, C5}: 2
{C1, C3}: 4
{C2, C5}: 2
{C2, C4}: 2
{C2, C3}: 4

We then generate a graph to do the ranking. If {C1, C2} is a pattern we assume there is an outgoing edge from C1 to C2. The graph in dictionary form:

{'C1': ['C2', 'C5', 'C3'], 'C2': ['C5', 'C4', 'C3']}

This shows that there are edges from C1 to C2, C5 and C3. Similarly, there are edges from C2 to C5, C4 and C3. We will use this graph to perform the ranking.

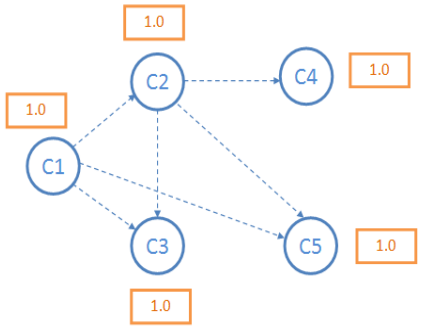

Figure 3: Step 1

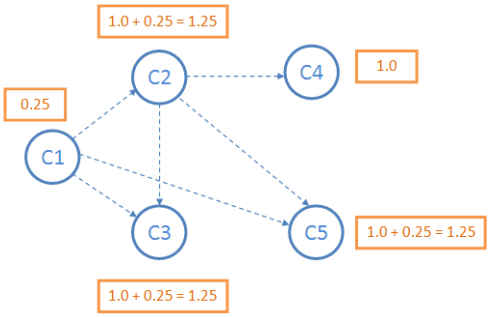

Figure 4: Step 2

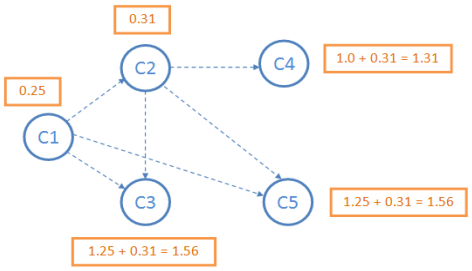

Figure 5: Step 3

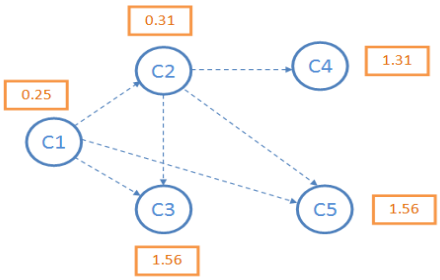

Figure 6: Step 4

Initially each page is assigned rank 1. As C1 has three outgoing edges, it divides 1 into four parts (3+1 = 4, ¼ = 0.25). It keeps 0.25 for itself and sends 0.25 each to its three immediate successors. Thus, pages C2, C3 and C5 have ranks 1.25. Again, C2 has three outgoing edges. So, it divides 1.25 into four parts, keeps 0.31 for itself and sends 0.31 to its immediate successors, C3, C4 and C5. C3 and C5 already have rank 1.25. Adding 0.31 to that they become 1.56. Rank of C4 becomes 1+0.31 = 1.31. So now we arrange companies in ascending order of ranks and generate recommendations. Thus, order of recommendations will be C3, C5, C4, C2, C1.

5. Testing

We wrote two automated scripts, **automate.py** and **k.py**. **automate.py** simulates the use of the engine by selecting a company at random from each category, performing sentiment analysis on its tweets and storing the sentiment. Though this is not a very accurate representation of the use of the engine, it gives us a dataset we can work off of. This script runs thrice a day. **k.py** recalculates **k** based on the direction of the shift. We then use this value to generate the investment scheme. It is calculated in the following way.

- Initially **k** is set to 1.0
- If the difference in sentiment is positive (i.e. we predict the stock price will rise) and it actually rises (i.e. our prediction is correct) we leave **k** undisturbed. The same applies for negative difference in sentiment.
- If our prediction goes wrong, we readjust **k**.
- If sentiment difference is negative and stock price goes up, we *increment* **k** by 0.1
- If sentiment difference is positive and stock price plummets, we *decrement* **k** by 0.1
- This script runs thrice a day

6. Conclusion and Future Work

We successfully extracted tweets and performed sentiment analysis on them. They give a fairly accurate guess of the direction of stock market shift. The recommendations generated based on clustering and ranking are quite sound based on our manual monitoring The entire processing takes a lot of time. So, for demonstration purposes we extract only 100 of the most recent tweets. But for increasing accuracy we need much more than that. So what could be done is distribute the set of tweets over parallel processors and combine the results at the end. Market sentiment is only one part of what decides rise and fall of stock prices. So in some cases our predictions can go wildly wrong. To improve upon this some Business Analytics have to be added, so that in combination with market sentiment can help the user make a more informed decision.

References

[1] Sentiment Analysis: http://en.wikipedia.org/wiki/Sentiment_analysis

[2] What is a Bull and a Bear market?http://content.moneyinstructor.com/693/what-bull-bear-market.html

[3] The Role of the Stock Market: http://www.updown.com/education/article/The-Role-of-the-Stock-Market

[4] Daily Market Summary: http://www.nasdaqtrader.com/Trader.aspxid=DailyMarketSummary

[5] Ray Chen, Marius Lazer, "Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement"

[6] SentiWordNet. An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. http://sentiwordnet.isti.cnr.it/

[7] Alexander Pak, Patrick Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining", Proceedings of the Seventh conference on International Language Resources and Evaluation LREC'10, Valletta, Malta, European Language Resources Association ELRA,(May 2010)

[8] Namrata Godbole, Manjunath Srinivasaiah, Steven Skiena, "Large-Scale Sentiment Analysis for News and Blogs", *Proceedings of the International Conference on Weblogs and Social Media ICWSM, (2007)*

[9] Efthymios Kouloumpis, Theresa Wilson, Johanna Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!", Proceedings of the I*nternational Conference on Weblogs and Social Media* ICWSM (2011).