R04

Social Investment Forecasting

Using Web Mining

By

Nikhil Berry and Mothusi Majinda

R04

Advised by

Prof. David Rossiter

Submitted in partial fulfillment

Of the requirements for COMP 4982

In the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2011-2012

Date of submission: April 19th, 2012

Table of Contents

List of Figures

List of Tables

Abstract

This report proposes an alternative approach to investment forecasting instead of the conventional technical analysis. This approach makes use of sentiment analysis of online news sources in addition to the sentiment analysis of social media sites, namely Twitter. These combined give users a strong sense of investors' expectations for different stocks. The system quantifies these expectations, calculating the weighted stock trend predictions. Users can view how the system predicted trends correlate to the actual market trends and adopt investment strategies accordingly. This investment forecasting system has been framed within an interactive and collaborative website. The website allows users to suggest words that improve the accuracy of the sentiment analysis. Furthermore, the social element of the website gives users a platform to exchange ideas and investment advice.

1. Introduction

1.1 Overview

Trading in stocks has always been lucrative if one can forecast the price trends of a particular stock. Given today's volatile markets, the need to more accurately forecast has never been greater. Numerous tools have been developed to monitor markets and trade using advanced algorithms, and some have had great success. However, as effective as they are, we propose another alternative. Stock prices can be heavily influenced by consumer expectations, and thus a change in stock price is oftentimes the final result of a change in consumer expectations. Instead of using tools to forecast prices given market data, we have developed a tool that utilizes text from various investment websites as a metric for measuring consumer expectations for a particular stock. If one could find a way to gauge the consumer expectations of a particular stock before the price begins to reflect these expectations, he would have a great advantage. The difficulty with this approach however, is finding sources where one can find timely, relevant and honest information from the market makers or analysts who truly understand the market.

The system we developed uses data from online social media [1], in particular Twitter [2]. Given the structure and reach of Twitter, which allows users to post brief comments of up to 140 characters, it can make an ideal source of live information about what is going on in the world if we can find Twitter members who are honest and understand the markets and market makers. Using the data from Twitter and employing web-mining techniques, we aimed to develop one component of our online system for predicting stock price trends. A second component uses daily financial reports, financial blogs and similar data mining techniques to produce alternate forecasts. A third component is a live feed of financial market data, allowing us to check if consumers' expectations match the actual market data.

For the sake of simplicity, we assumed that all sources of information are unbiased, honest, knowledgeable and representative of all market makers' actions and intentions. We also assume that no governments are able to manipulate markets for political reasons.

1.2 Objectives

The following are the objectives for the project:

1. Access several financial news sources and extract useful content.
2. Use the Twitter API to access tweets relating to stocks.
3. Clean our data to shorten the processing time.
4. Interpret tweets and financial news sources to come to a prediction.
5. Pass predictions to a website which has graphical representations of trends.

1.3 Literature Survey

Stock prediction using web mining is a very common area of research. We examined two approaches.

Financial Prediction using Web Mining Approaches – Ma Yao [3]

The focus of this thesis is to use available text information as opposed to numerical information in order to forecast stock prices. This approach relies on investors' emotions and utilizes a large amount of available text information rather than technical analysis. It not only identifies market behavior from this text information, but it helps investors understand short-term market behavior.

Yao identified several limitations and areas for improvement for his system. One area of research we will expand upon is related to text information sources. While his choice of financial sources gives relevant daily data, the relevance fades as the day progresses. Our system will use live text information from social media websites like Twitter, which will be able to give relevant information at any given point in time. In addition to this, we will also implement a prediction method for financial sources similar to Yao's, allowing us to compare the effectiveness of our stock prediction given our two different data sources.

Stock Market Prediction Based on Public Attentions: a Social Web Mining Approach - Ailun Yi [4]

Yi chose three sources: Newswire, Blog and Twitter. He uses three days of Twitter data for the training data. This leads to the system losing accuracy outside of the sampling period. To improve upon this, Yi suggested that future systems attempt to include long-term history and utilize prior knowledge. Yi also suggested finding a way of measuring whether there is a sufficient level of information available about the

given stock or company we would like to predict stock prices for. This would allow us to know whether the system can generate any worthwhile conclusions based on the level of information available, thereby allowing us to make more accurate conclusions about the effectiveness of the system.

2. Design

The project is divided into the following key components:

1. The **Link extractor** reads in root URLs from a text file. For each root URL, the link extractor accesses the webpage and extracts every URL from that webpage. All extracted URLs are saved to one file to be processed by the text extractor.
2. The **Google** component reads the list of stocks from a text file. For each stock, a customized Google search is conducted and the URLs are saved to another text file to be processed by text extractor.
3. The **Text extractor** reads the text file created by the link extractor and Google component. For each URL in the text file, the text extractor creates a text file with the webpage content of that particular URL.
4. The **Twitter program** reads from a text file containing our target stocks. For each stock, the twitter program runs a query using the Twitter Search API. The program then creates a text file containing the tweets for each query.
5. The **Tagger program** goes through all of the text files created by the text extractor and the twitter program and removes irrelevant words. This is achieved by using the Stanford POS Tagger. The tagger program then overwrites the text files without the irrelevant words.
6. After tagging, each text file is processed using the **Sentiment analysis program**. This uses a database of positive and negative words to assign weights to each article and Twitter tweet containing the stocks the system is looking for. The program then calculates an aggregate weight for each stock using the weights from the articles and the Twitter tweets. This serves as the prediction for each stock.
7. A **Web interface** allows the user to pick stocks and see the forecasted results. The interface also includes administrative controls to modify the input parameters and other variables. Users can also specify which stocks they want information about.
8. The **Email** component sends the users a daily email containing the predictions of the system.
9. The **Variance** component calculates the variance and the standard deviation of the predictions from different sources for all the stocks.

10. The **Price** component gets the latest change in prices for all the stocks in the system.

The System runs every day before the NASDAQ market opens. Users can see the results in an interactive manner through the website.

System Design Diagram

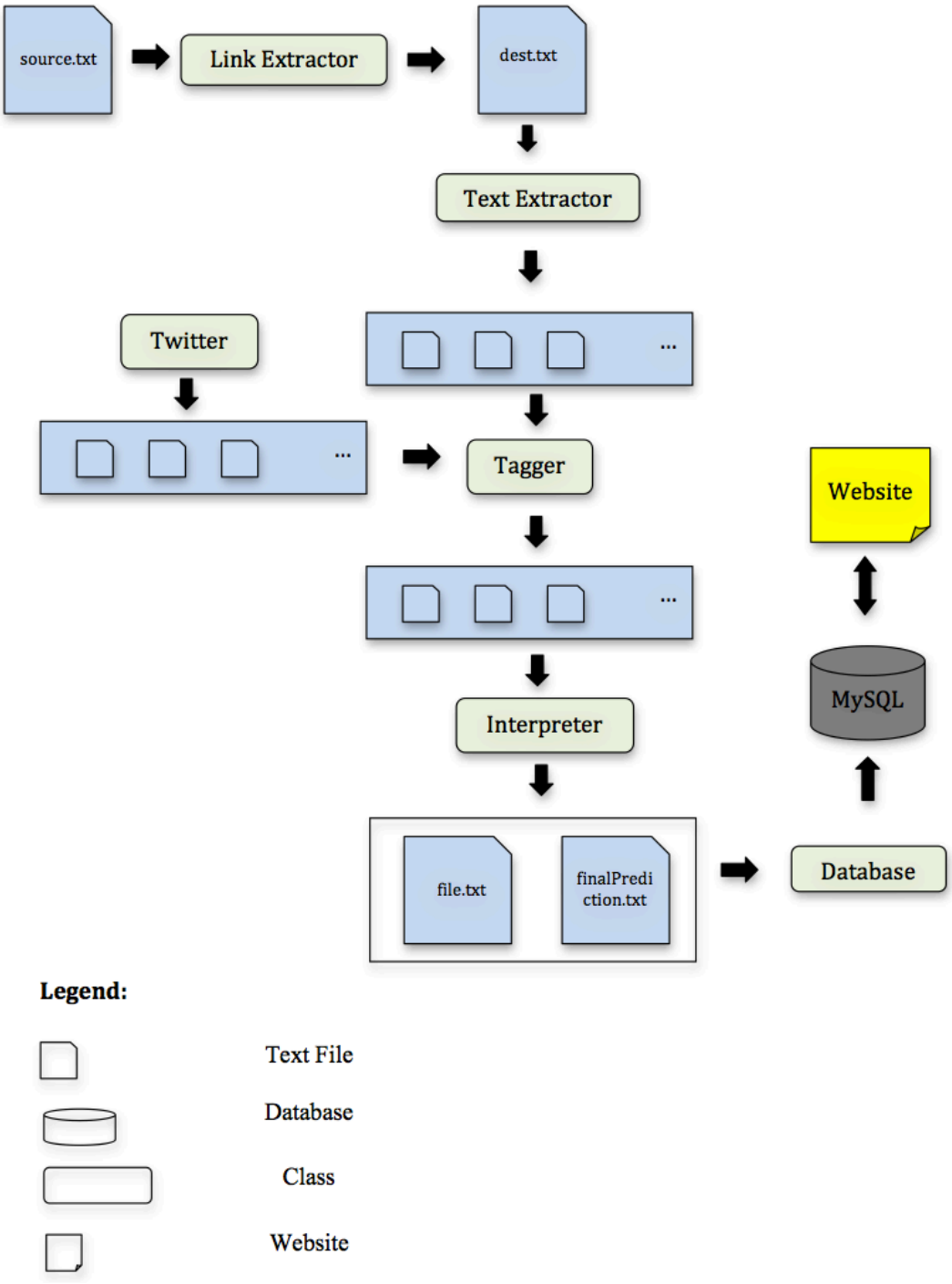


Figure 1: System Design Diagram

3. Implementation

In this project, the following components have been implemented:

- Link Extractor
- Google Component
- Twitter Program
- Text Extractor
- Tagger Program
- Sentiment Analysis
- Email Component
- Variance Component
- Price Component
- Website

Link Extractor

The link extractor component has been implemented using Java and the HTML parser library. This component reads the target URLs from a text file (**source.txt**). In addition, the number of levels to be searched can be specified. Then, for each of the URLs in source.txt, the system retrieves all of the URLs in the webpage and in the specified levels. These are written to a file known as **dest.txt**.

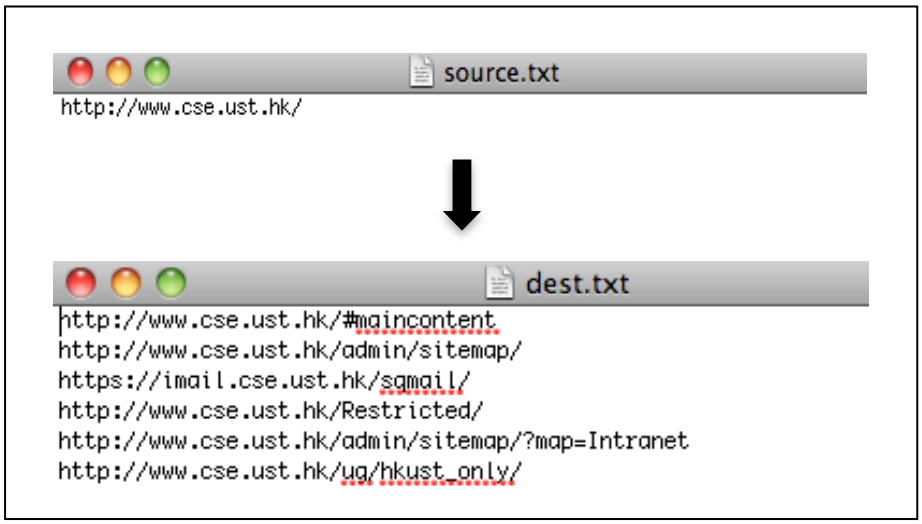


Figure 2 - Parts of the text files

Google Component

The **Google Component** reads the list of stocks from the stocks text file. Each stock is appended to an upward or downward sentiment word and a Google search is conducted. A parameter that limits the results to only pages that have been updated in the last 24 hours is specified. The system takes the top 10 results, append them to a text file (dest.txt) and send them to the text extractor component.

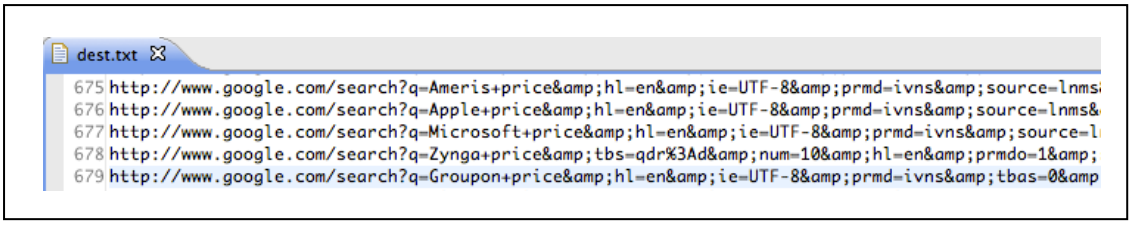


Figure 3 - Google links in dest.txt

Twitter

The twitter program has been implemented using Java and the twitter4j library. Using this library, the system constructs a URL, which utilizes the Twitter search API. The Twitter search API then returns a JSON file, which the system then parses.

Example URL

http://search.twitter.com/search.json?q=oil&rpp=100&include_entities=true&result_type=mixed

Parameters

- q: Search query
- rpp: The number of tweets to return per page
- result_type: Specifies type of search results

Screenshot of the returned results

{"completed_in":0.052,"max_id":167968175009435648,"max_id_str":"167968175009435648","next_page":"?
page=2&max_id=167968175009435648&q=oil&rpp=100&include_entities=1","page":1,"query":"oil","refresh_url":"?
since_id=167968175009435648&q=oil&include_entities=1","results":[{"created_at":"Thu, 09 Feb 2012 10:44:24 +000
[],"urls":[{"url":"http://t.co/NF4Rnsom","expanded_url":"http://www.bostonglobe.com/sports/2012/02/09/oil-can-
pitching-with-red-sox/ndAb0cYV5J8nTc3JYrA0sJ/story.html","display_url":"bostonglobe.com/sports/2012/02\u2026",
[67,87]}],"user_mentions":[]},"from_user":"Buster_ESPN","from_user_id":88763317,"from_user_id_str":"88763317",
Olney","geo":null,"id":167559467079180288,"id_str":"167559467079180288","iso_language_code":"en","metadata":
{"recent_retweets":5,"result_type":"popular"},"profile_image_url":"http://a2.twimg.com/profile_images/518275190
le_image_url_https":"https://si0.twimg.com/profile_images/518275190/olney_buster_m_normal.jpg","source":"<a
href="http://twitter.com/">web","text":"Oil Can Boyd says he pitched under the influenc
http://t.co/NF4Rnsom","to_user":null,"to_user_id":null,"to_user_id_str":null,"to_user_name":null},{"created_at
+0000","entities":{"hashtags":[],"urls":[],"user_mentions":
[]},"from_user":"sportsguy33","from_user_id":32765534,"from_user_id_str":"32765534","from_user_name":"Bill
Simmons","geo":null,"id":167429389343137792,"id_str":"167429389343137792","iso_language_code":"en","metadata":

Figure 4 - Twitter Program parameters and results

The constructor of the class takes the search query as input and creates text files, which use the search query as the file name. Each tweet is saved on a new line in the text file. The text files are saved in a folder, which has the current date as its name.

The twitter program is executed for each of the stocks in the system. The text files containing the tweets are then sent to the tagger component.

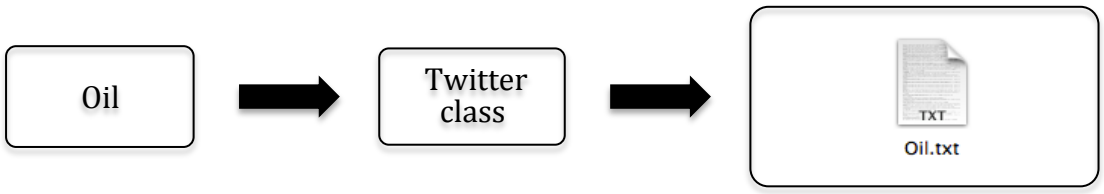


Figure 5 - Twitter Program flow

Text Extractor

The text extractor component was implemented using Java and uses the HTML parser library. This component reads the target links from a text file (dest.txt). Then each of the links in dest.txt is visited. The HTML tags are parsed and the text content of the webpage is extracted. The text from the source webpage is then written to another file, which is given the same name as the link.

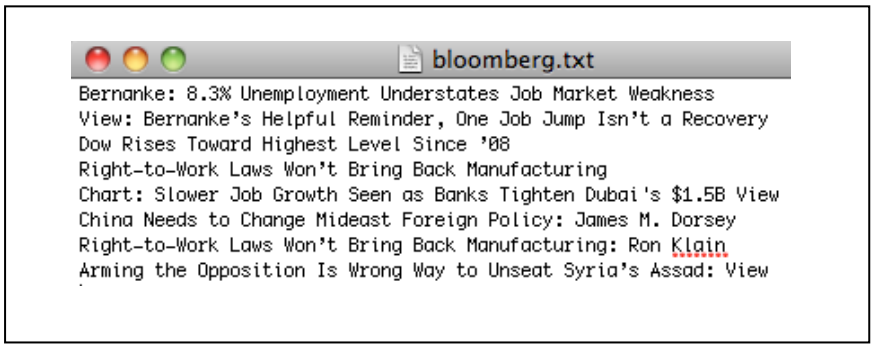


Figure 6 – Part of text obtained from www.bloomberg.com

Tagger

The Tagger program was implemented using Java and makes use of the Stanford POS Tagger. The Tagger program takes a text file as input and passes it to the Stanford POS Tagger, which tags each word according to its usage type (noun, verb, adjective, etc.). The Tagger program then removes words that are not necessary for the purpose of text analysis and then overwrites the text file with this new string.

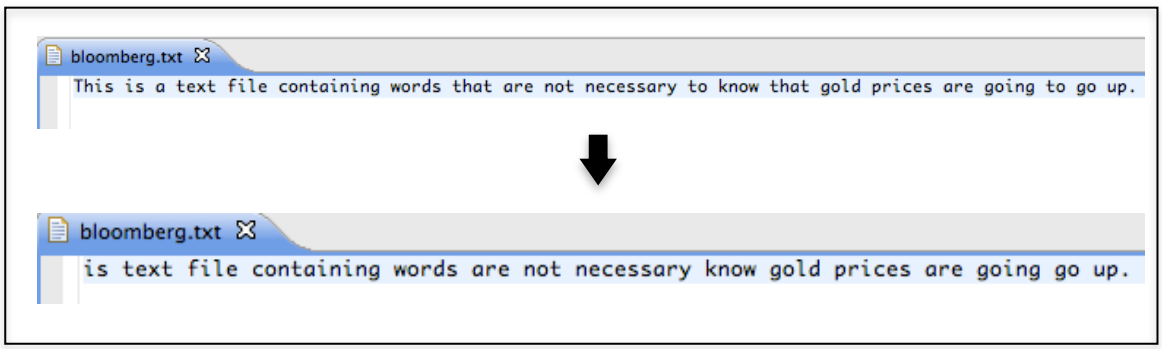


Figure 7 - Sample text file before and after Tagger processing

The Tagger currently keeps:

- JJR Adjective, comparative
- JJS Adjective, superlative
- NN Noun, singular or mass
- NNS Noun, plural
- NNP Proper noun, singular
- NNPS Proper noun, plural
- RBR Adverb, comparative
- RBS Adverb, superlative
- VB Verb, base form
- VBD Verb, past tense
- VBG Verb, gerund or present participle
- VBN Verb, past participle
- VBP Verb, non-3rd person singular present
- VBZ Verb, 3rd person singular present

In addition, the tagger writes to a file the words that are tagged as adjectives, adverbs and verbs. This file is used in the Collaborate component of the website.

Sentiment Analysis

For this component, there are two classes. The first one is **Generic Tree** class, which contains functions the system uses to represent an article as a tree. The second is **Tree** class, which makes use of Generic Tree class to represent the article as a Tree. The root of the tree is the entire article. The next level down is each sentence of the article. The last level is every word in each sentence.

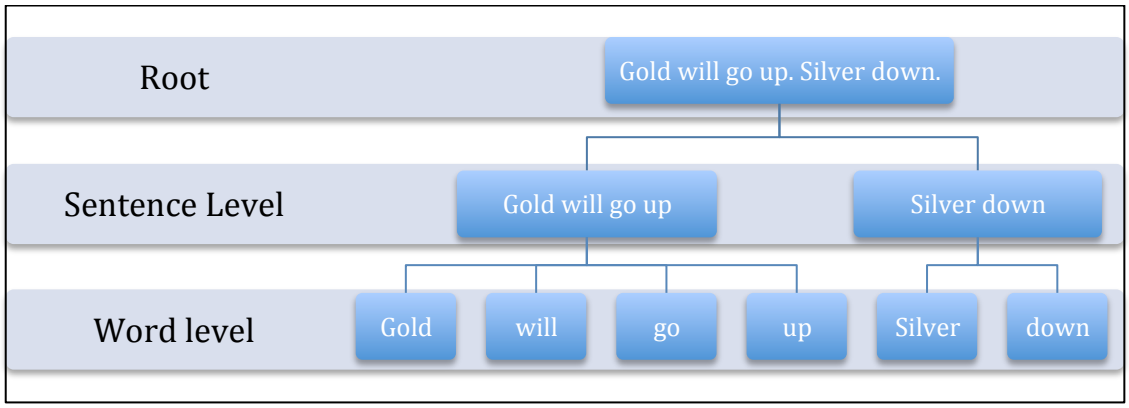


Figure 8 - Example tree structure

The resource word lists are grouped into three categories, namely, positive sentiment word list, negative sentiment word list and the negation word list (See Appendix). Liang Xun and Rong-Chang Chen [5] employed a similar technique for their sentiment analysis.

For each sentence node of the tree, there are two array lists. The first array list, known as **stock**, contains the list of stocks found in the sentence. The second, known as **weight**, contains the weight associated with each stock in the sentence. The system assumes that a sentence ends with a full stop followed by a space. This assumption is necessary because it allows us to not take into account full stops used in acronyms.

Markets will open at 9 a.m. They should shoot up.

The article above will be split into two sentences and not three.

The weight at a particular index in the weight array list corresponds to the stock at the same index in the stock array list. If a sentence contains both a stock and a positive or a negative word, then the system assigns a weight of 1.0 or -1.0 respectively.

Gold will go up. Silver should go down.

In the above article, Gold is assigned a weight of +1.0 and silver -1.0.

If a sentence contains only a positive or negative word, but no stock, then we assume that the sentence is related to the stock found in the previous sentence and assign a relatively lower weight.

Gold has been in the news a lot lately. I think it will go down.

In the above article, Gold will be assigned a negative weight whose absolute value will be less than 1. We use this strategy because the further the positive or negative word is from the stock name, the lower its weight.

I don't know what to do with gold.

If the sentence has only a stock name, but no positive or negative weight, then we assign a weight of 0.0, and it will not be included in the final aggregation of weights. In the sentence above, Gold is assigned a weight of 0.0.

We also assume that for a positive or negative word to be associated with a stock in a given sentence, it must be positioned within **seven** words of the stock. This is because if the distance between the stock and the positive or negative word increases beyond a certain threshold, the positive or negative word may not be related to the stock mentioned in the sentence.

This is a sentence showing that although I'm talking about gold, I could also mention that I am going to buy a car.

In the above sentence, the system would not assign a positive weight to gold because "buy" is too far away from "gold".

You shouldn't buy Gold.

The system also checks for words, which can negate the meaning of a positive or negative word. In the sentence above, even though “buy” is a positive word, the presence of a negation word before it reverses the actual meaning of the sentence. The system checks for the negation word up to a maximum of seven words from the occurrence of a positive or negative word. Therefore, Gold will be assigned a weight of -1.0 and not +1.0.

The system first calculates the weights associated with all the stocks found in a particular article and writes the result to a text file (**file.txt**). The system processes all the articles in this way, and the results are appended to file.txt. Similarly, all of the Tweets are passed to the Sentiment Analysis component, and the weights are assigned and appended to the file.txt. After all the articles have been processed, the system aggregates all the weights for a particular stock in file.txt and writes it to a new file, known as **finalPrediction.txt.** The finalPrediction.txt file contains the final prediction for each of the stocks found by the system.

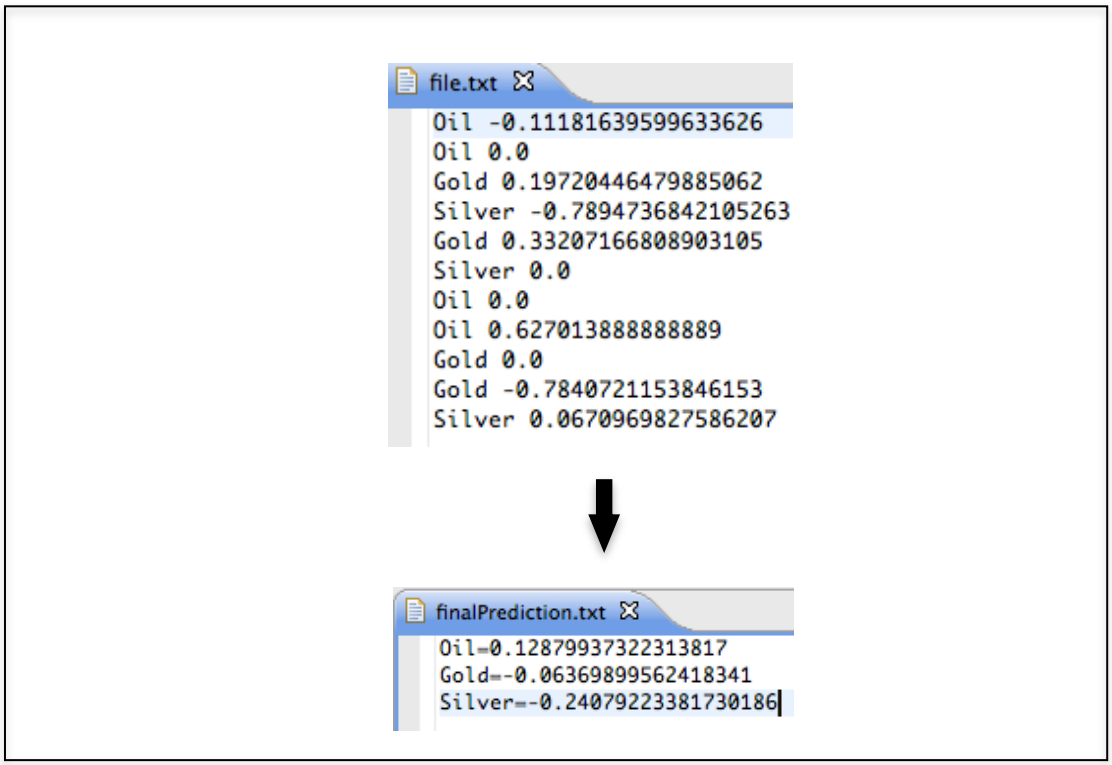


Figure 9 - Contents of file.txt and finalPrediction.txt after a sample run.

In the next step, records are read from the finalPrediction.txt file and inserted into the Predictions MySQL database. The website uses the information from this database to display a graphical interpretation of the predicted trend.

The system has a minimum requirement on the information required to make a prediction for a particular stock. That is to say, if the system finds only one article or source, which makes a prediction on a particular stock, the system will use this one source as the only prediction. To address this issue, the system disregards the prediction for a particular stock if the number of sources/articles found is less than a minimum threshold requirement.

In addition, for each individual article we used the TFIDF (Term Frequency Inverse Document Frequency) [6]. If the TFIDF weight is below a certain threshold, we disregard the article in our calculations. TFIDF is a "statistical measure used to evaluate how important a term (i.e., word, feature, etc.) is to a message in a corpus. The importance increases proportionally to the number of times the term appears in the message but it is offset by the frequency of the term in the corpus."

$$TFIDF(w) = TF(w) \times IDF(w)$$

$$TF(w) = \frac{n(w)}{\sum_{w'} n(w')}$$

$$IDF(w) = log(\frac{|M|}{\{m : w \in m\}})$$

M is the set of all messages while n(w) is the frequency of the term w in a message. m is the number of documents where the term w appears.

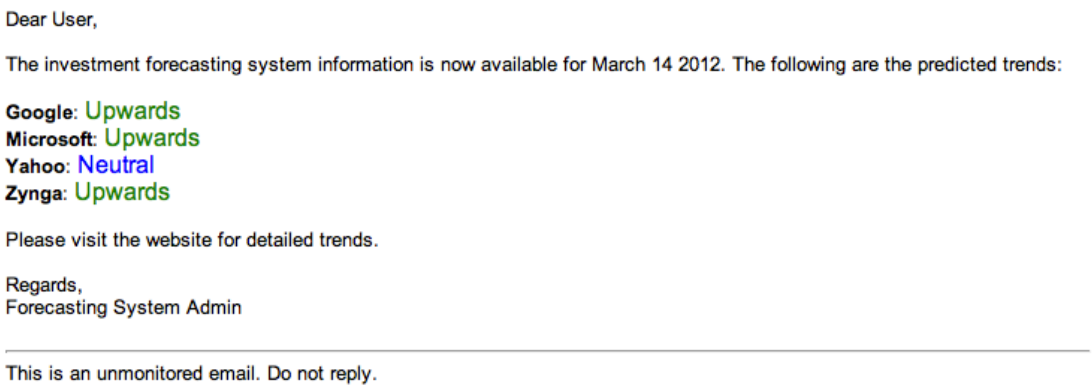
To implement TFIDF in the system, we disregard a stock prediction for that particular day if the number of opinions is below a certain threshold.

Email Component

The email component sends the users a daily email containing the predictions of the system. The users can select from the website if they want to receive a daily email from the system. A list of all the users who wish to receive daily emails is retrieved from the Users database. This component makes use of the Java mail library, which makes use of the SMTP server [7] to send emails. Each user can specify which stocks they want information about. The emails are color coded as follows:

Sentiment Type	Color
Strong Upward Sentiment	Dark Green
Medium Upward Sentiment	Green
Neutral	Blue
Medium Downward Sentiment	Red
Strong Downward Sentiment	Dark Red

Table 1: Email color code


Dear User,

The investment forecasting system information is now available for March 14 2012. The following are the predicted trends:

Google: Upwards
Microsoft: Upwards
Yahoo: Neutral
Zynga: Upwards

Please visit the website for detailed trends.

Regards,
Forecasting System Admin

This is an unmonitored email. Do not reply.

Figure 10 - Daily Email

Variance Component

The Variance component is used to calculate the variance and standard deviation of a particular stock's prediction. This component helps the users to infer whether the predictions from different sources align with the final prediction from the system for that particular stock. In other words, it shows the degree to which opinions about a particular stock vary from the system's final prediction.

As the variance of a particular stock approaches zero, the opinions about that stock become more consistent with each other, thus, making the system's final prediction

more reliable. Inversely, if the variance is large, it means there are inconsistent opinions about that stock and therefore makes the final prediction less reliable.

The calculated variance and standard deviation for each stock in the system is inserted into the Statistics database.

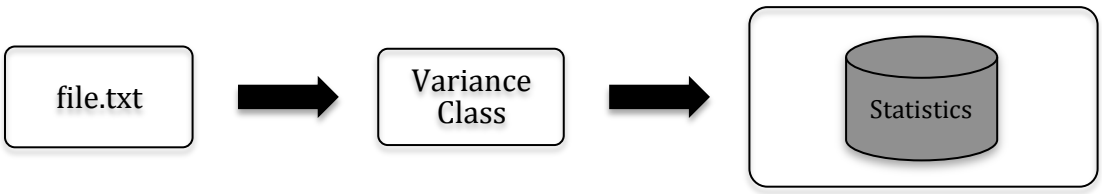


Figure 11 - Variance

Price Component

The price component is used to fetch prices from the Yahoo API for the stocks in the system. For each stock in the system, the percentage change in prices is fetched and stored in the database. The data obtained by this component helps us in the evaluation of the system. A customized URL is constructed for each of the stocks in the system and results obtained are extracted from the webpage using the text extractor component. The prices and the changes for all the stocks in the system are inserted into the Prices database.

```
http://download.finance.yahoo.com/d/quotes.csv?s=GOOG&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=MSFT&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=YHOO&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=ZNGA&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=XNPT&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=SDBT&f=p2
http://download.finance.yahoo.com/d/quotes.csv?s=ABCB&f=p2
```

Figure 12 - Yahoo API URLs

Website

The website has been implemented using an array of programming languages, including HTML, JavaScript, jQuery, PHP and SQL.

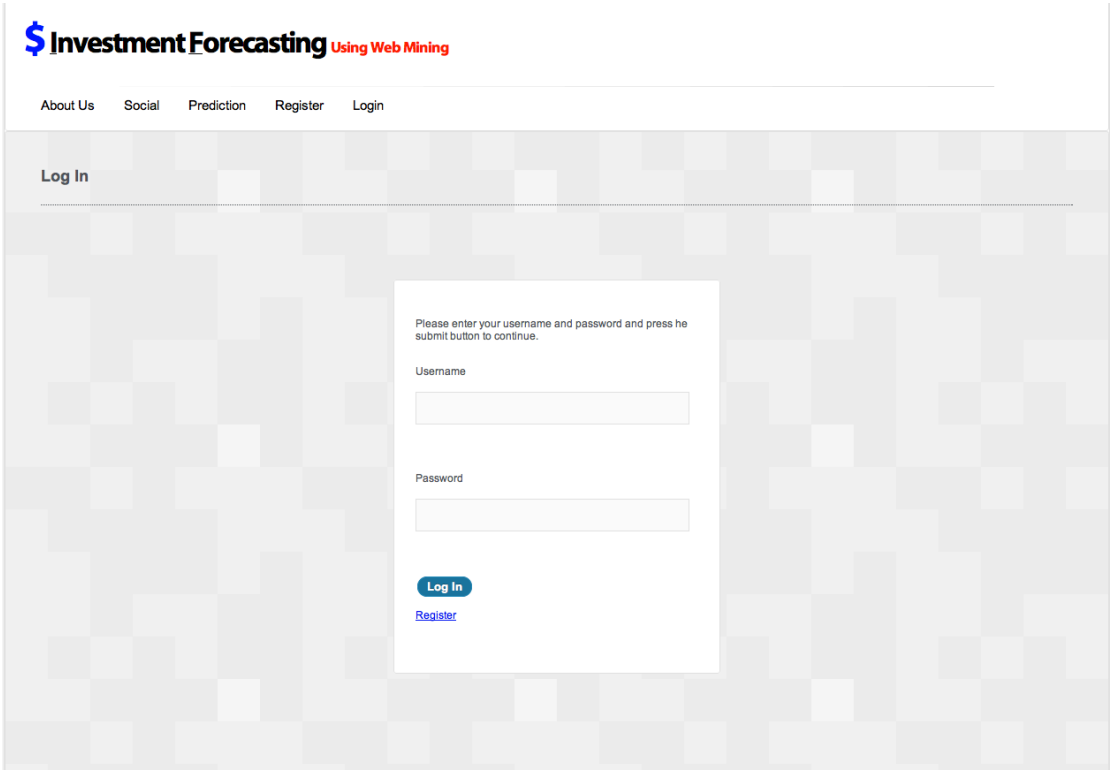


Figure 13 - Website homepage

As a new user, one would click the register link to go to the registration page. A continuing user would simply enter their Username and Password and click login to proceed into the website.

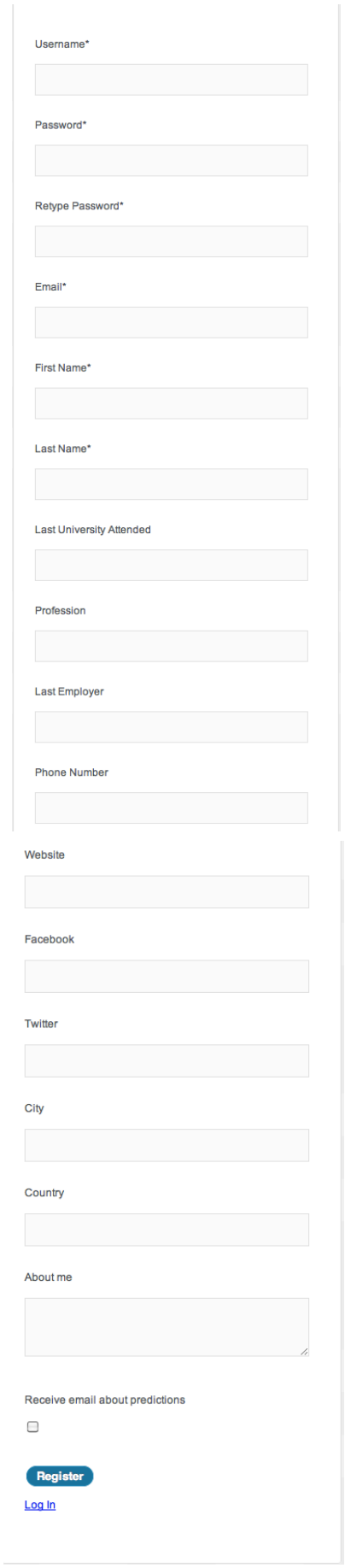


Figure 14 – The registration page

The registration page requires a new user to enter their Username, Password, Email, First name and Last name. The user will only be referred to by their Username on the website, thereby allowing complete anonymity for users in sensitive financial positions. The other fields (University, Profession, Employer, etc.) are used to fill out the user profile, but are not mandatory fields for the registration process. Once you have completed the registration, you may now login.

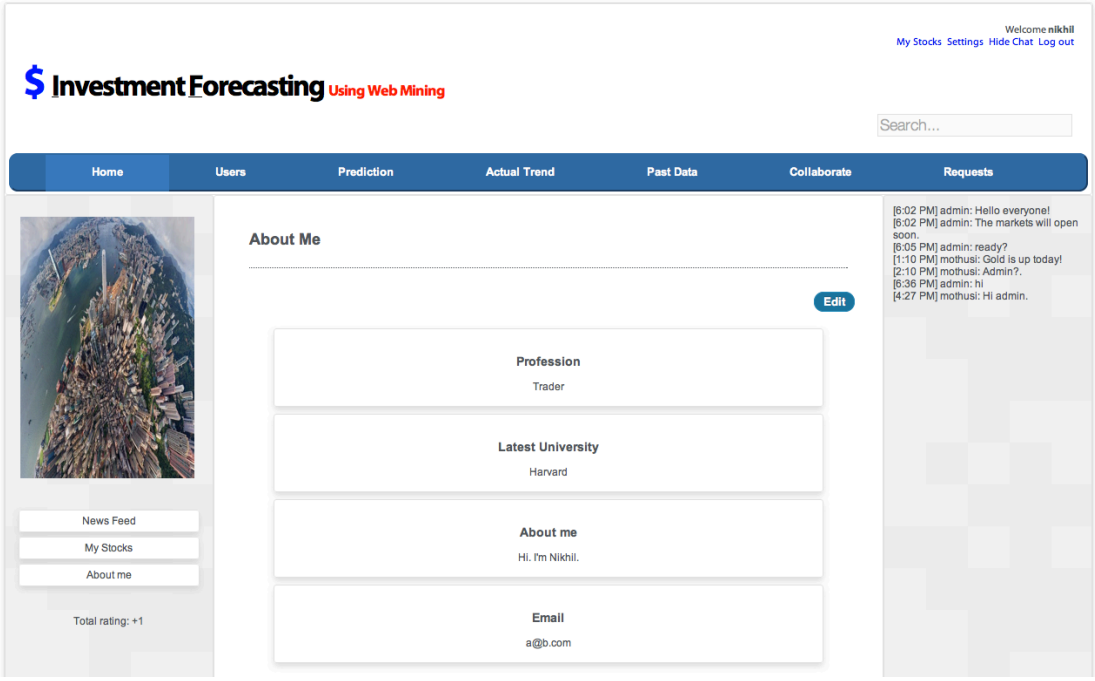


Figure 15 – The user homepage

Once a user logins, they are directed to the user homepage shown in Figure 15. This allows the user to retrieve a variety of stock information, modify their profile, interact with other users and participate in the collaborative improvement of the system's effectiveness and accuracy.

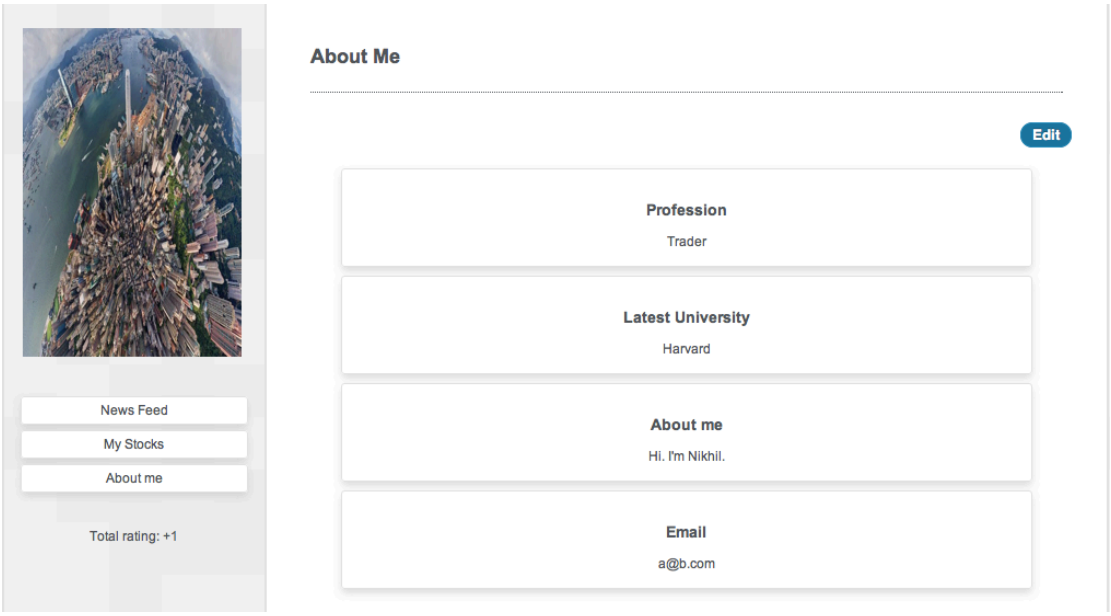


Figure 16 – About me

Every user has an about me section as shown in Figure 16. This displays all public information of the user. The Profession, Employer and About Me sections are fields that could potentially indicate the reliability of a given user. Each user can modify, remove and add information to their profile including their profile picture.

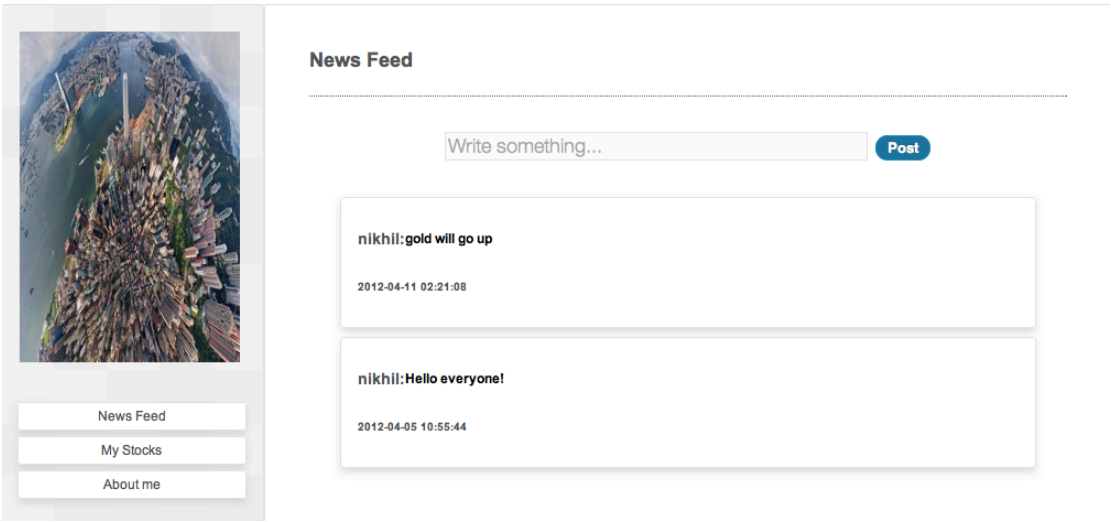


Figure 17 – The News Feed

The news feed serves as a means for users to post their predictions about particular stocks. All news feed posts are public, meaning any user can view any user's past posts.

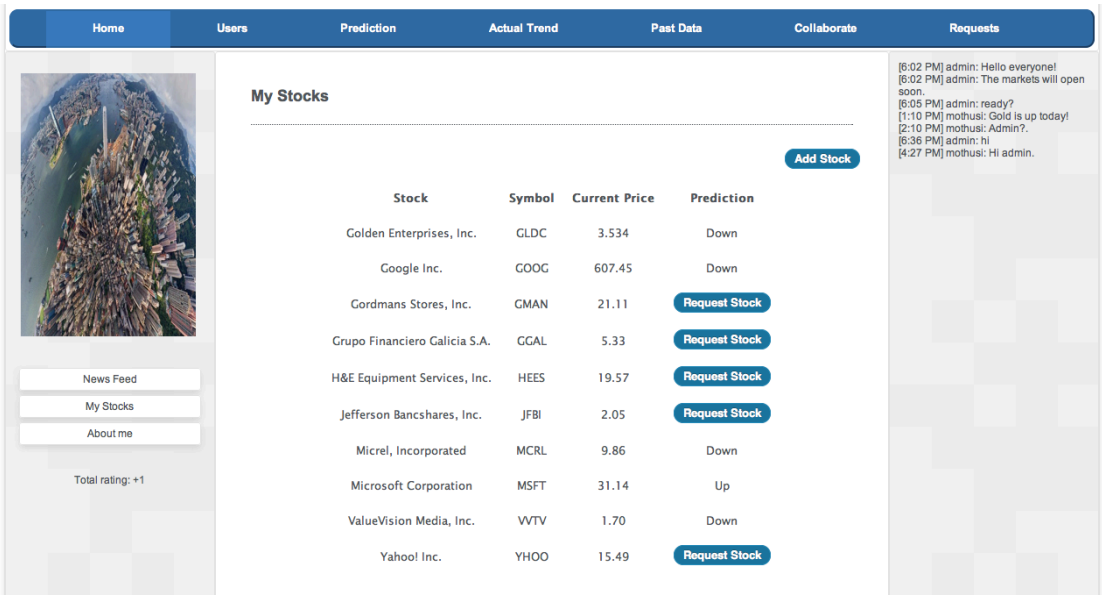


Figure 18 – My Stocks

Each user can choose a list of stocks they would like to follow by using the My Stocks page. My stocks page shows the current price of the stock, as well as the current system prediction for that stock. If the stock is currently not being analyzed by the system, a user can make a request to have the stock added to the system.

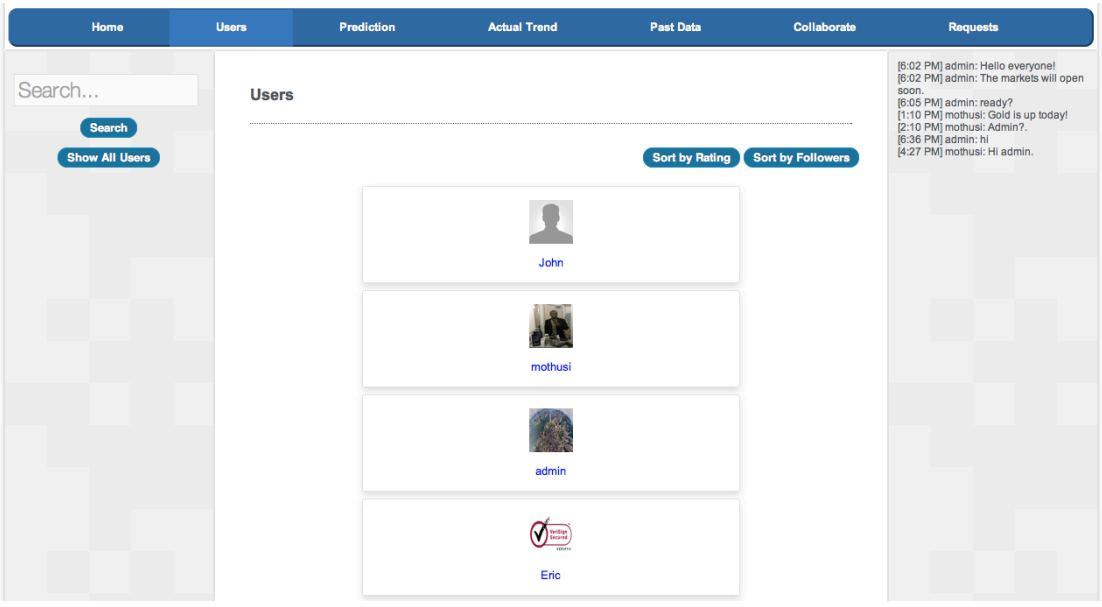


Figure 19 – Searching for Users

The system allows users to search for other users using their username, or simply by scrolling through a list of all users. Additionally, the users can be sorted by rating and number of followers (discussed in detail in later sections).

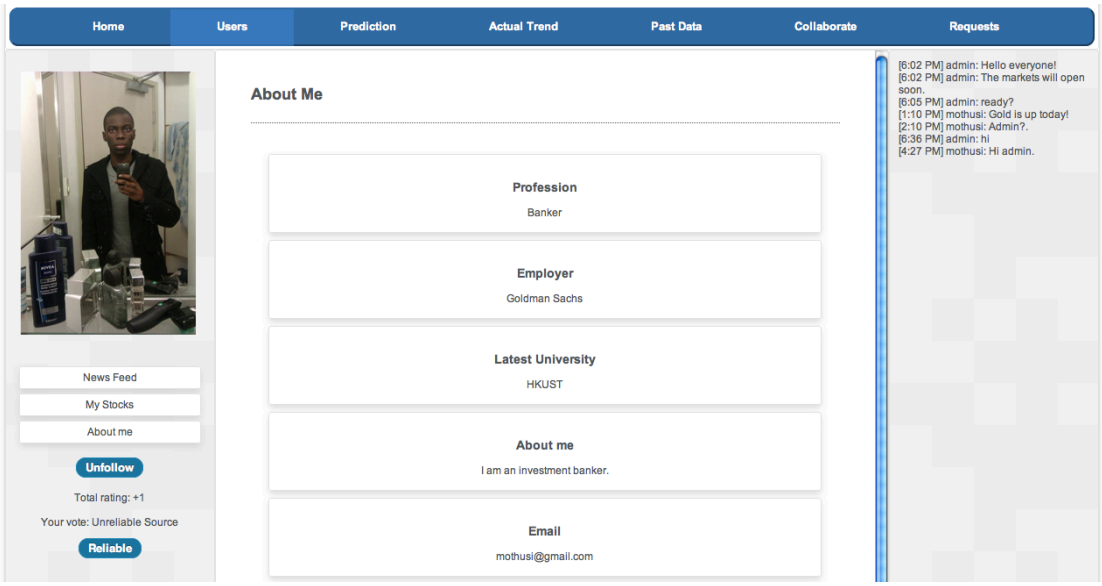


Figure 20 – User Mothusi's profile

Once you choose a user, you are directed to their profile. Their profile will show that particular user's News Feed, About Me, and their My Stocks. If you find a user reliable, or unreliable, you may vote for them using the Reliable and Unreliable buttons.

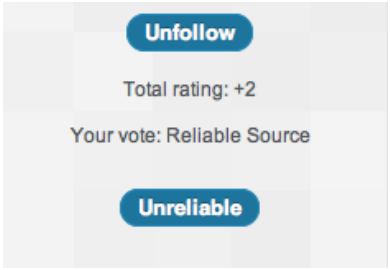


Figure 21 – Follow and Reliability

If you value this particular user's predictions, you may choose to follow them (Fig. 21).

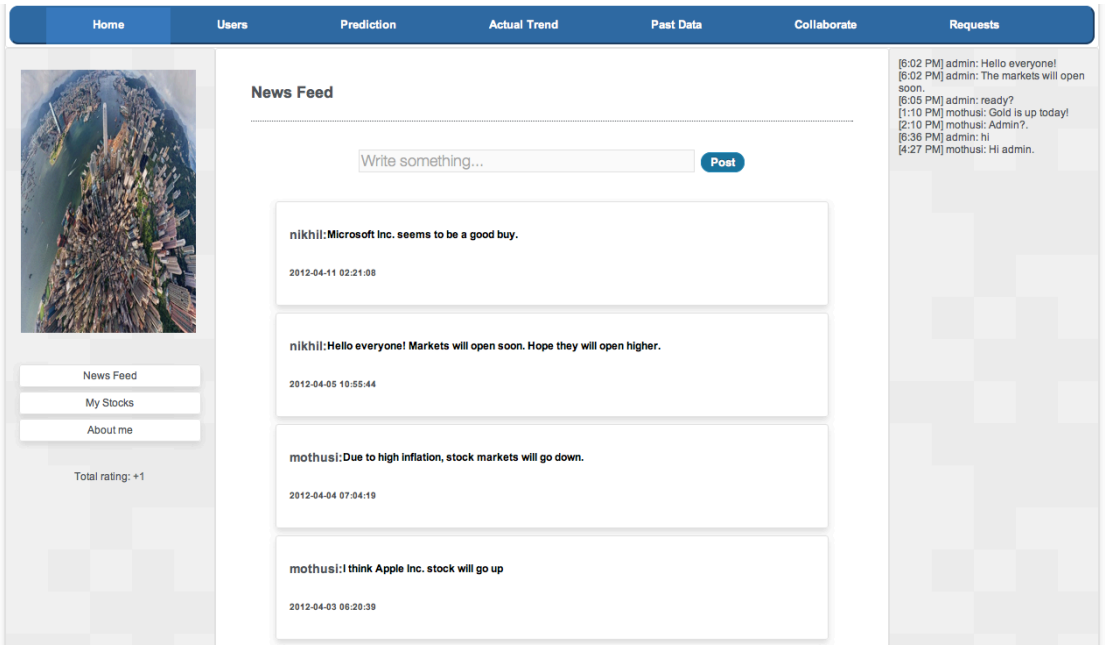


Figure 22 – User Nikhil following user mothusi

When you follow a particular user, their news feed posts appear on your own news feed (Fig. 22), meaning you do not have to navigate to their page to view their posts. Since all posts are public, the system has been intentionally developed in a way that allows users to vote on the reliability of a user without necessarily following them. This allows greater flexibility for users, allowing them to vote a user unreliable without having to be cluttered by their potentially inaccurate predictions in their news feed. Each user on the system has a total number of followers and total rating. As mentioned in the user search section above, you may search for users with more followers and higher rating, which generally means this user is more reliable or more trusted by the community. A user's total rating is the sum of all their ratings, where a vote of unreliable is -1 and a vote of reliable is 1.

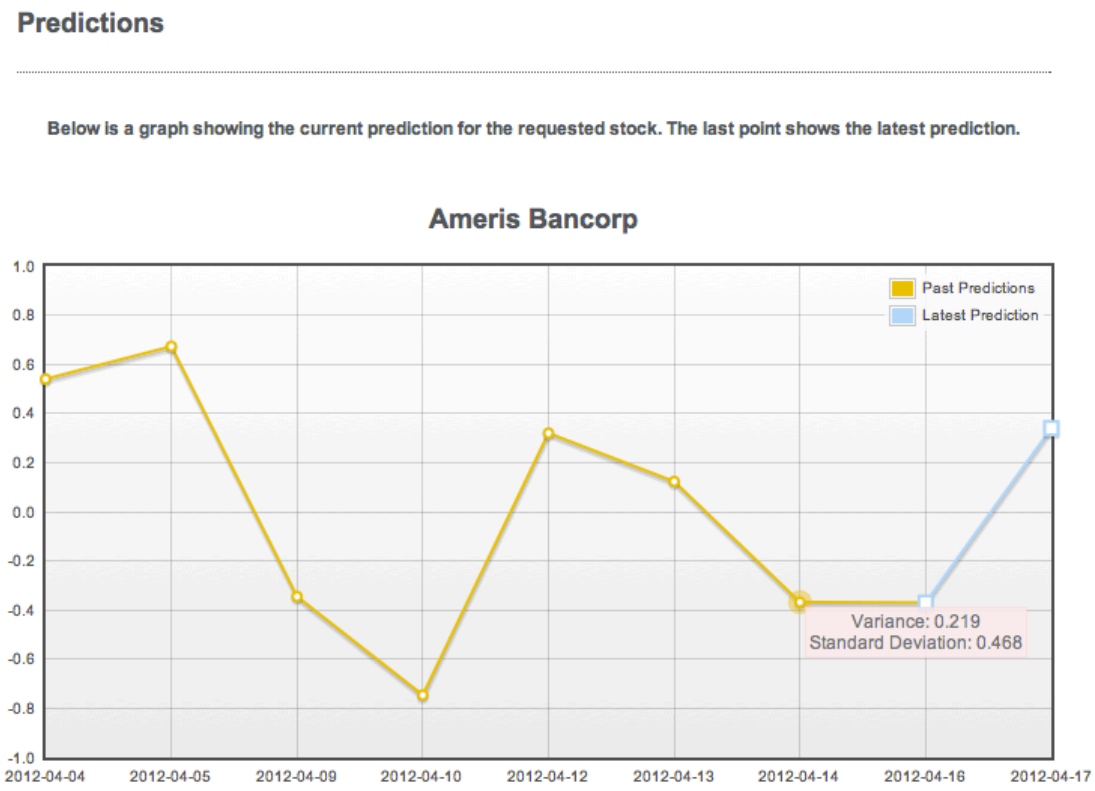


Figure 23 – System predictions for Ameris Bancorp

Every user of the system can access the system predictions for all the stocks currently being processed by the system. A user chooses the stock from the drop down, and may choose a date range, and the graph is generated showing the trend and prediction. The last point is the last system generated prediction, meaning it indicates the prediction of the system for the next market day. The user can hover over each point in the graph to view the variance and standard deviation for that particular point.

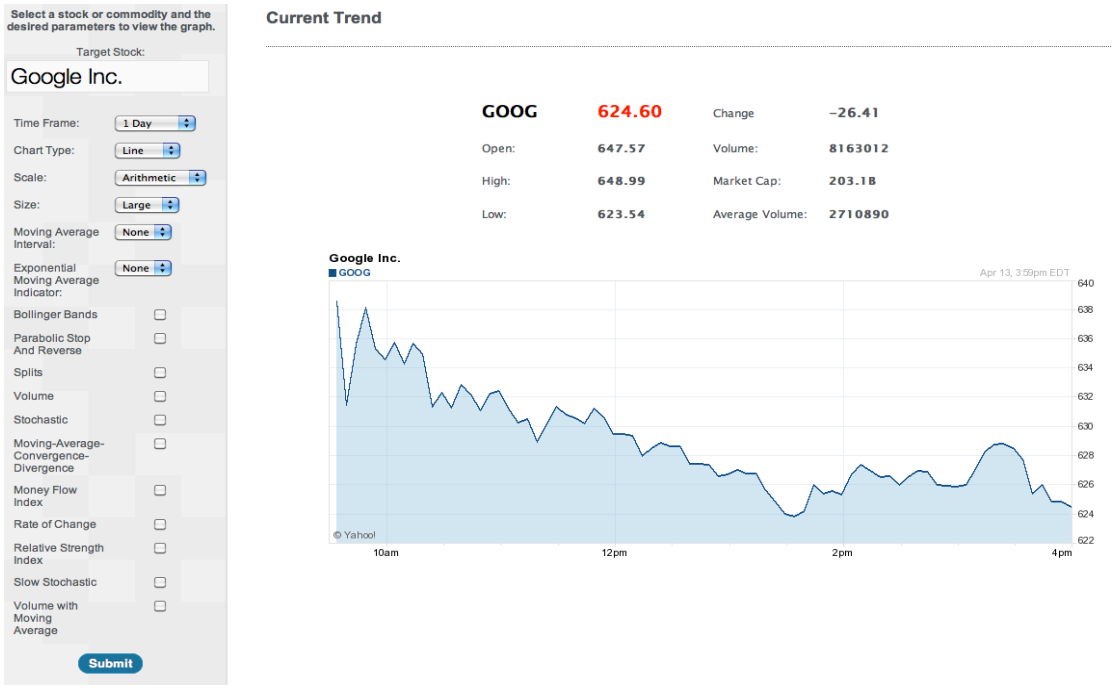


Figure 24 –Actual market data for Google Inc.

A user can also view the actual trend of a particular stock (Fig 24). This has been implemented using the Yahoo API. When markets are open, the page shows real time prices changes in the particular stock price.

A user can also view the actual trend for a particular stock and compare it to the system prediction. This is done by using the search bar located in the top right of every webpage. The search bar has a drop down of several thousand stocks listed on the NASDAQ. To allow us to map the system predictions to actual market data, we have chosen to limit the stocks only to those listed on NASDAQ.

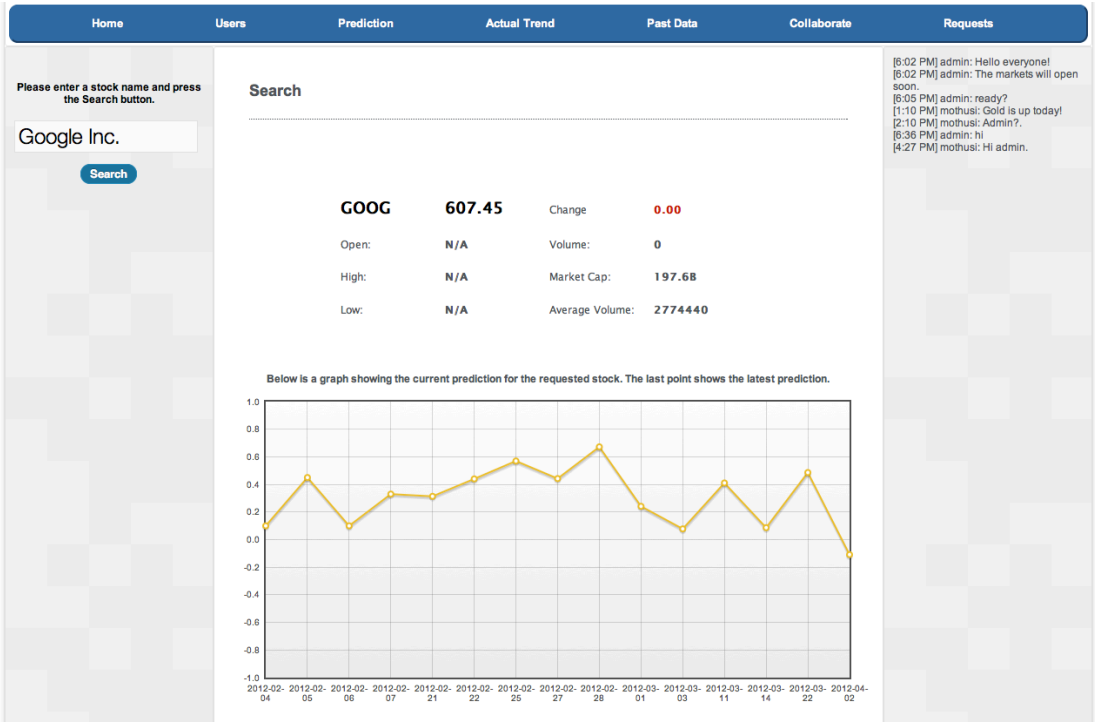


Figure 25 – Google market data when markets are closed compared to System prediction

The search bar would redirect a user to a page that shows stock price information and also the system generated predictions for that particular stock (Fig. 25).

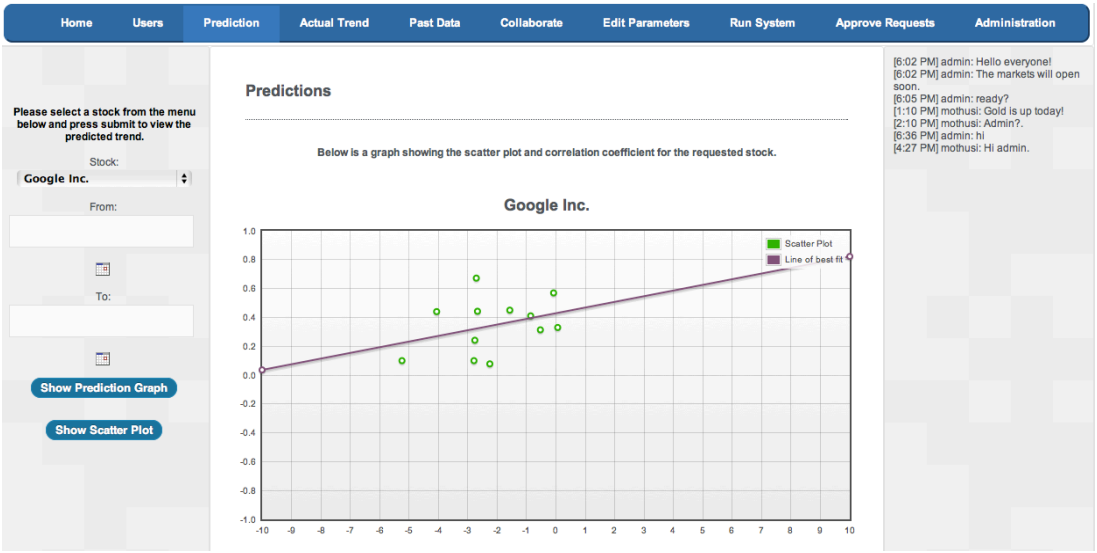


Figure 26 – Scatterplot data for Google Inc.

As part of the system evaluation, a scatterplot component has been implemented. This shows the correlation between the system's predicted stock trend and the actual stock trend. For each day, the variance and standard deviation are also calculated, which are

used to show the degree to which individual sources deviate from the system's final predicted trend. The line of best fit and the correlation coefficient for each stock are calculated as further statistical analysis of the system.

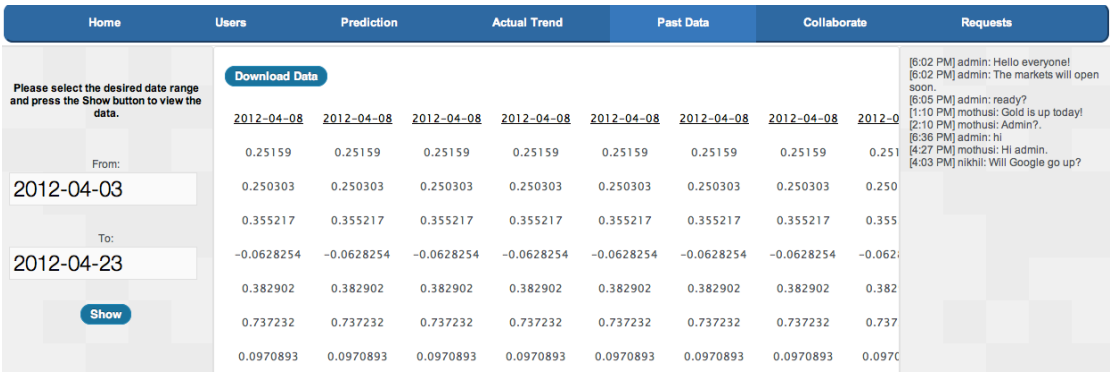


Figure 27 – Past Data

Users can also view the past data for all the stocks in the system by specifying a date range. In addition, users can download the data in excel format and use it for their own purpose.

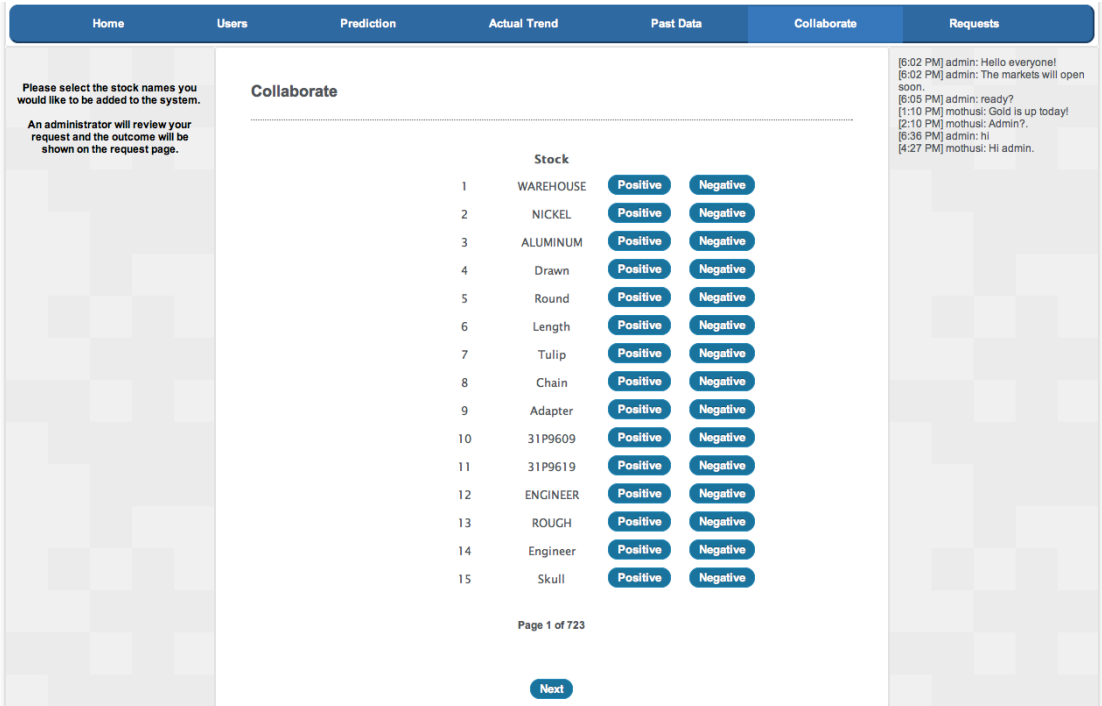


Figure 28 – Collaborate

To allow the user community to collaboratively improve the system performance, there is a component called Collaborate. The Collaborate component allows users to assist administrators in the improvement of the system by suggesting possible positive and negative sentiment indicator words. The list of words is generated during the sentiment analysis phase and a text dump is created with all the possible adverbs, adjectives and verbs. When a user makes a request to add a positive or negative word, their request is sent to an administrator for approval.

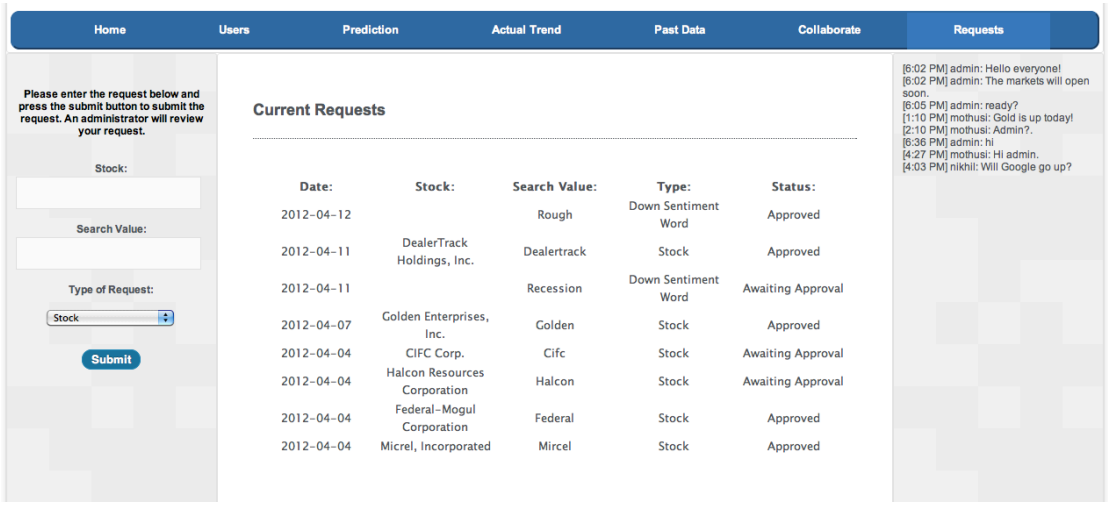


Figure 29 – User Requests

A log of all user requests is kept in the request page. In addition to suggesting positive or negative words, the user can also suggest new sources, new negation words (not, shouldn't etc.) as well as new stocks to follow. All such requests require administrator approval. If however, the logged in user is currently an administrator or root level user (discussed in detail later), all requests are immediately approved and a log is stored in the request database.

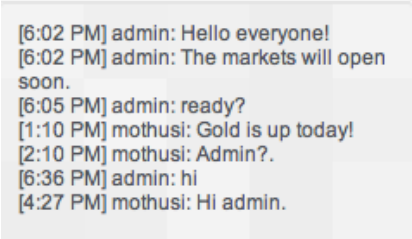


Figure 30 – User Chat

A chat feature has also been implemented that allows users to converse and interact. This gives users a real-time method to communicate and share ideas about certain stocks.

In addition to the feature available to the standard users, administrator and root administrator accounts have several additional features.

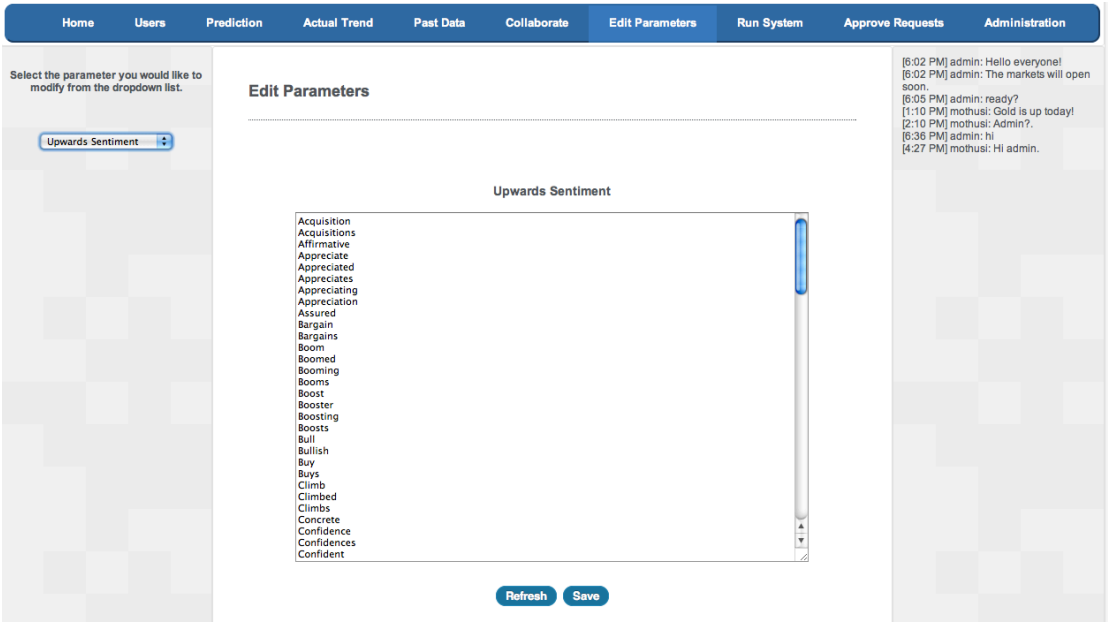


Figure 31 – Edit Parameters

Both administrator and root accounts can edit the parameters used by the system for text analysis. This includes upwards sentiment, downwards sentiment, sources, negation words and stocks.

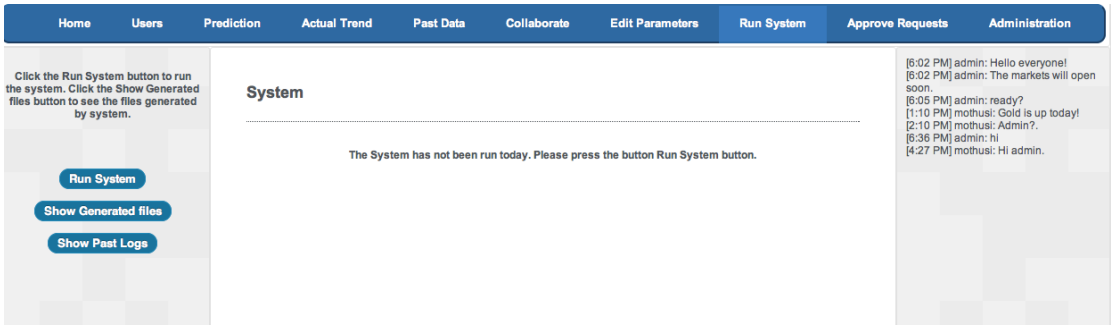


Figure 32 – Run System

Both administrator types can run the system ad hoc via the website interface. Although the system is currently executed by a cron job every day before the market opens, the administrator may wish to run the system ad hoc. In order to remove the possibility of accidental double execution of the system, there is a feature, which allows the administrator to know whether the system has been run today. If the administrator chooses to run the system, they can see the real time file generation by

the system via the Show Generated files button. All logs generated by the system are kept, which the administrators can browse via the Show Past Logs button.

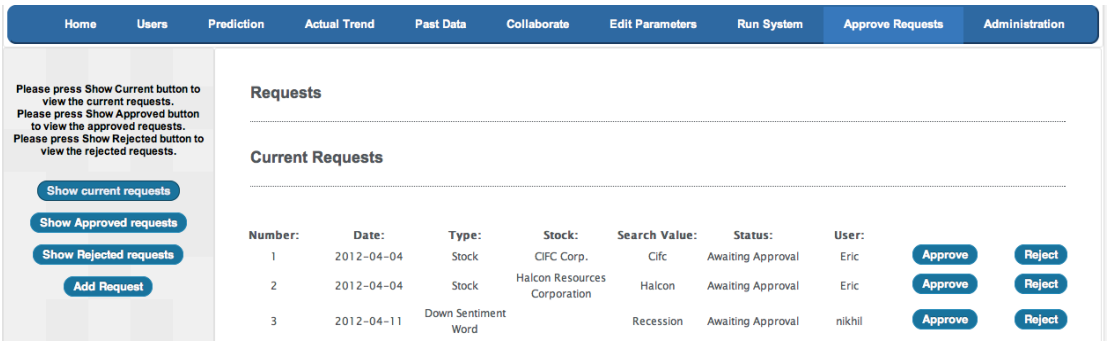


Figure 33 – Requests

In addition to being able to add stocks, positive sentiment, negative sentiment, sources and negation words without the need of administrator approval, the administrator can approve requests made by users on the system. These are shown by the Show current requests button. For a list of all approved and rejected requests, the administrator has an interface to view both.

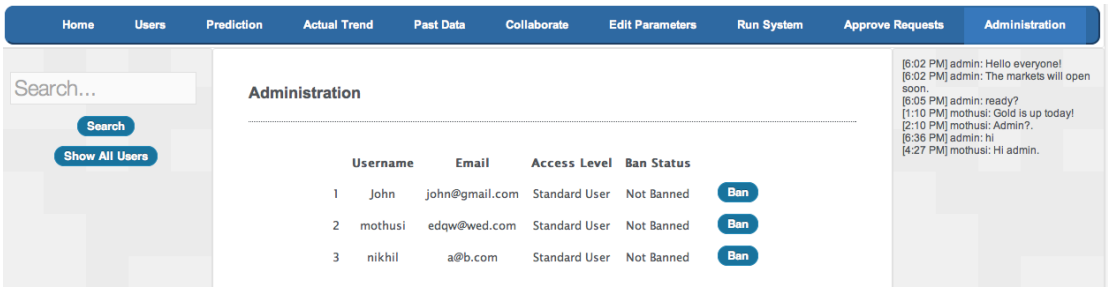


Figure 34 – Standard administrator Administration interface

The area that administrators and the root user differ is the Administration interface. A normal administrator can ban and unban any user except other administrators and the root user. (Fig. 34)

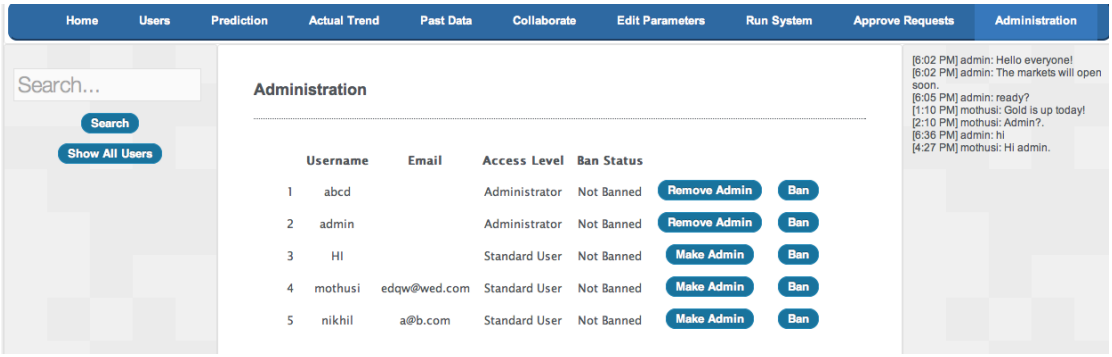


Figure 35– Root administrator Administration interface

The root, however, can ban and unban *any* user and can also add and remove administrator access to users in the system. (Fig. 35) The rationale behind this is to limit the possibility of accidental bans of administrators and eliminate the need to ever have to manually enter the database to modify access levels if things go wrong. With this implementation, the root administrator can repair any access issues since they are not ban-able and cannot have their access removed.

4. Testing

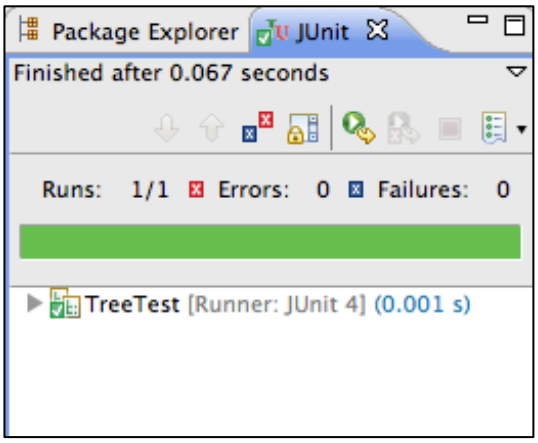


Figure 36 – a JUnit test case in eclipse

The testing phase involved finding errors in the system and debugging the system. For testing the various components of the system, we used **Unit Testing**. Drivers and stubs were used wherever required. Next, testing was carried out to check if the components interact correctly and if their functionality is as expected, also known as **Integration Testing**. Last but not the least, all the components were tested as a whole, also known as **System Testing.** In essence, we took a bottom-up approach to testing, testing the system incrementally as it increased in components and complexity.

Testing the link extractor

To test the link extractor component, we chose a webpage and manually compiled all of the URLs in the webpage. We then ran the link extractor using the webpage as the input and compared the results with the expected results from our manual compilation. The expected and actual results were found to be the same, and therefore we assumed that this component works as expected.

Testing the twitter program

To test the twitter program, we manually constructed the Twitter search API URL and saved the results in a text file. Next, we ran the Twitter program with the same search query and parameters and saved the results to a text file. Then, the content of the two text files was compared and it was found to be the same.

Testing the text extractor

To test the text extractor, a webpage was chosen and the HTML source code was saved to a text file. Then, the HTML tag information was manually removed. Using the same webpage as input, the text extractor was executed. Next, a simple text comparison algorithm was implemented in Java, which compared two text files after removing all whitespaces. The content of the two text files was found to be the same. Therefore, we concluded that the text extractor works as expected.

Testing the tagger

To test the Tagger, we created a JUnit test that took the input and output for the Tagger and asserted that the length of the string of the output is shorter. We passed the stocks and the positive and negative words in the input string as parameters to the JUnit test. It then used them to check if the output string still contained the necessary stocks and positive or negative words needed for text analysis after being processed by Tagger.

Testing sentiment analysis

To test the sentiment analysis, we created a JUnit test suite that tested each individual component of the text analysis. This included tests for multiple stocks per sentence, weight assignment, sentence splitting, multiple positive and negative words in a sentence, positive or negative words distant from the subject stock, and weight negation.

In addition, a survey was sent to selected people. In the survey, the selected people had to judge the sentiment in the given articles. The results are analyzed in the evaluation section.

Testing the system

To test the system as a whole, the actual trend and the predicted trend from the system were compared. The accuracy and the correlation for selected stocks were computed. In addition, the variance of predictions from different sources for the selected stocks was computed. This would help us to analyze how much the opinions from different sources vary from the system's final prediction. The results are summarized in the evaluation section.

5. Evaluation

5.1 Correlation between Actual and Predicted trends

The system was run daily starting from 1st February 2012 until 10th April 2012 for the purpose of evaluation. Similarly, actual trend data was collected for 4 stocks for the same time period and then a comparison was drawn. The comparison set contains 48 data points. The actual trend is represented on the X Axis and the predicted trend on the Y Axis.

The formula for calculating the change in actual trend is as follows:

$$Change\ \% = \left(\frac{Closing\ price - Open\ Price}{Open\ Price}\right) * 100$$

The line of best fit has been calculated using linear regression (method of least squares). The formula for the same is as follows:

$$Line\ of\ Best\ fit\ equation: y = w_0 + w_1.x$$

Let x_m be the mean of the predictor values of the Dataset D, and y_m the mean of the response values in D

$$w_1 = \frac{\sum_{i=1}^{i=|D|}(x_i - x_m)(y_i - y_m)}{\sum_{i=1}^{i=|D|}(x_i - x_m)^2}$$

$$w_0 = y_m - w_1.x$$

The Line of best fit minimizes the sums of squared errors (SSE) [8]. SSE is calculated using the following formula:

$$SSE = \sum_{i=1}^{i=|D|}(Actual\ value - Predicted\ value)^2$$

The correlation coefficient helps to evaluate how the actual and predicted trend move in relation to each other. The correlation coefficient has been calculated using the following formula:

$$r = \frac{\sum_{i=1}^{i=|D|}(x_i - x_m)(y_i - y_m)}{|D|.\sigma_x.\sigma_y}$$

Where σ_x is the standard deviation of x variable and σ_y is the standard deviation of y variable.

The following are the scatter plots, line of best fit and correlation coefficients for the selected stocks:

Apple Inc. (AAPL)

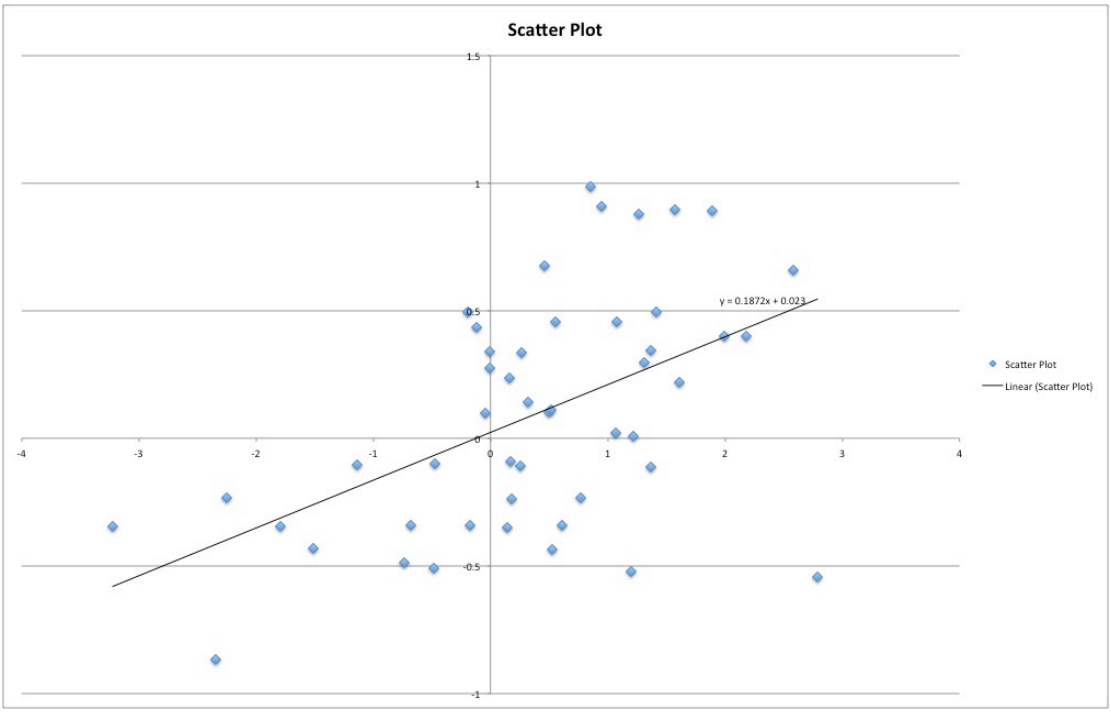


Figure 37 - Scatter Plot for Apple Inc.

- Correlation between the actual and predicted trend: 0.510085749
- Line of best fit equation: y = 0.1872x + 0.023

Ameris Bancorp (ABCB)

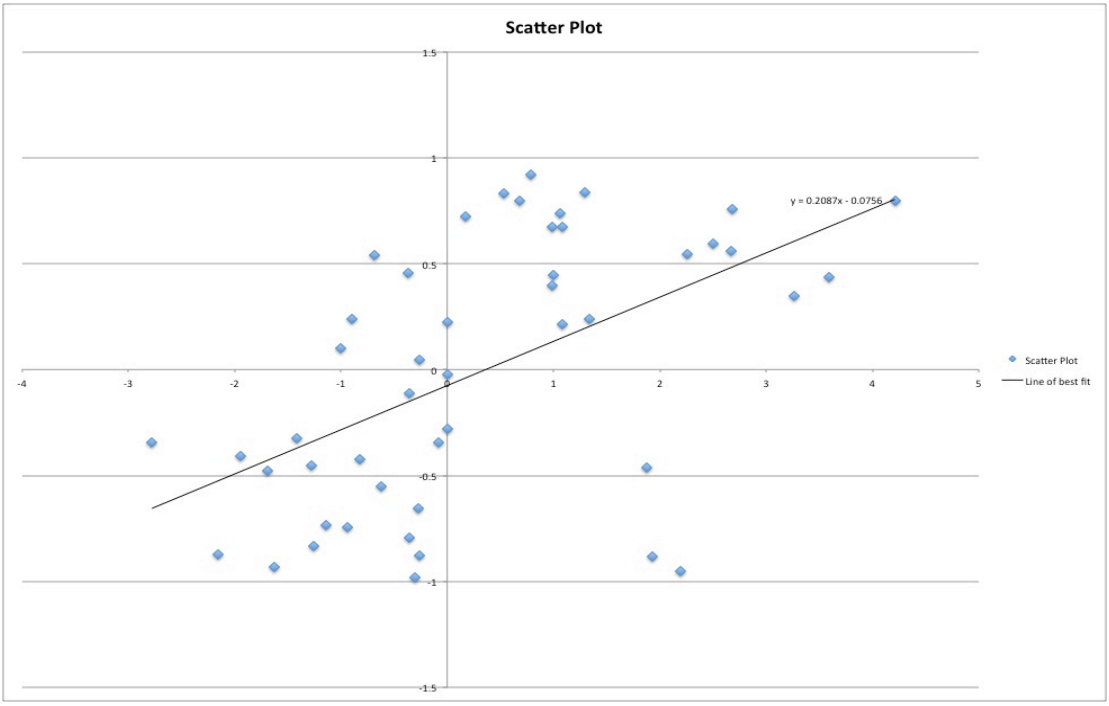


Figure 38 - Scatter Plot for Ameris Bancorp

- Correlation between the actual and predicted trend: 0.534558791
- Line of best fit equation: y = 0.2087x + 0.0756

Microsoft Corporation (MSFT)

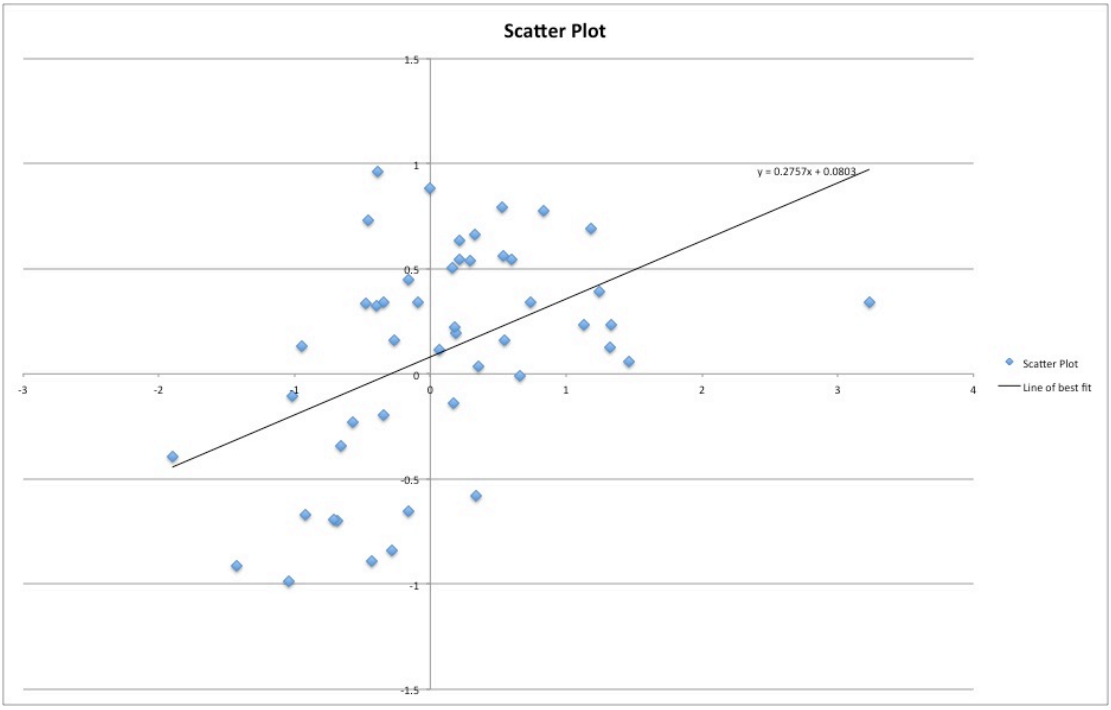


Figure 39 - Scatter Plot for Microsoft Corporation

- Correlation between the actual and predicted trend: 0.461043882
- Line of best fit equation: y = 0.2757x + 0.0803

Federal-Mogul Corp. (FDML)

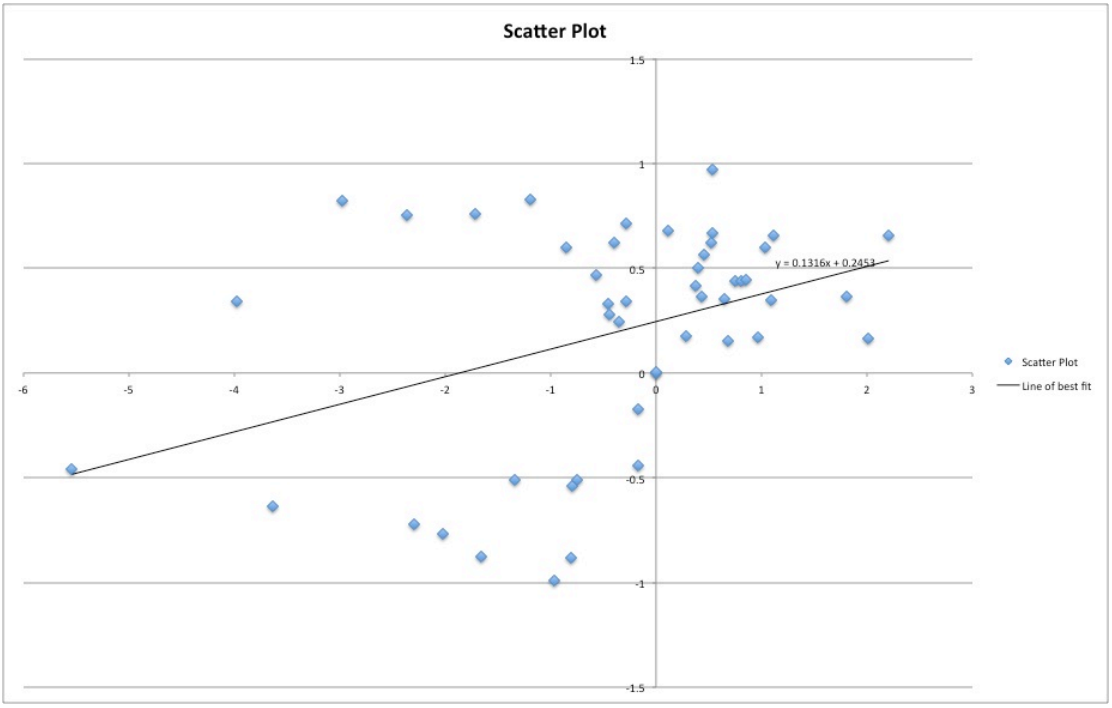


Figure 40 - Scatter Plot for Federal-Mogul Corp

- Correlation between the actual and predicted trend: 0.645199049
- Line of best fit equation: y = 0.1316x + 0.2453

5.2 Precision and Recall

For measuring the accuracy of the system, we used the Precision and Recall methods. These methods help us evaluate if the actual trend change aligns with the predicted trend change and therefore, help us to calculate the accuracy of the predictions. The accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's actual (true) value. In other words, it helps us to evaluate if the actual and predicted trend moved in the same direction.

"Precision is the fraction of retrieved instances that are relevant, while **Recall** is the fraction of relevant instances that are retrieved. A high **recall** means nothing has been missed but a lot of useless results may have sifted through (which would imply low **precision**). High **precision** means that everything returned was a relevant result, but not all the relevant items might have been found (which would imply low **recall**)" [9].

Actual class		
Predicted Class	tp (True Positive)	fp (False Positive)
	fn (False Negative)	tn (True Negative)

Table 2: Precision and Recall

- Precision = $\frac{tp}{tp + fp}$
- Recall = $\frac{tp}{tp + fn}$
- Accuracy = $\frac{tp + tn}{tp + tn + fp + fn}$

Microsoft Corporation (MSFT)

Total entries in the dataset: 47

True Positive	True Negative	False Positive	False Negative
22	13	9	3

Table 3: Microsoft Corporation

$$Precision = \frac{22}{(22 + 9)} = 0.709$$

$$Recall = \frac{22}{(22 + 3)} = 0.880$$

$$Accuracy = \frac{35}{47} = 0.744$$

Federal-Mogul Corp. (FDML)

Total entries in the dataset: 46

True Positive	True Negative	False Positive	False Negative
19	12	13	2

Table 4: Federal-Mogul Corp.

$$Precision = \frac{19}{(19 + 13)} = 0.593$$

$$Recall = \frac{19}{(19 + 2)} = 0.904$$

$$Accuracy = \frac{31}{46} = 0.673$$

Ameris Bancorp (ABCB)

Total entries in the dataset: 45

True Positive	True Negative	False Positive	False Negative
19	18	5	3

Table 5: Ameris Bancorp

$$Precision = \frac{19}{(19 + 5)} = 0.791$$

$$Recall = \frac{19}{(19 + 3)} = 0.904$$

$$Accuracy = \frac{31}{46} = 0.863$$

Apple Inc. (AAPL)

Total entries in the dataset: 48

True Positive	True Negative	False Positive	False Negative
22	11	5	10

Table 6: Apple Inc.

$$Precision = \frac{22}{(22+5)} = 0.814$$

$$Recall = \frac{22}{(22+10)} = 0.687$$

$$Accuracy = \frac{33}{48} = 0.687$$

5.3 Variance analysis

The variance is a statistical measure to show how far data is spread out. The variance is utilized to show how much the opinions vary from different sources. Since the system's predictions are in the range of -1 to 1, this means the variance ranges from 0 to 2. A variance of 0 indicates strong cohesion of opinions within the sources whereas a large variance shows a lack of consistent opinion between sources. The variance can therefore be used to judge how close the opinions from different sources are to the final system prediction for a particular stock. The following is the formula for calculating variance:

$$s_{N-1}^2 = \frac{1}{N-1}\sum_{i=1}^{n}(x_i - x_m)^2$$

Where x_m is the sample mean and x_i is the ith data point and N is the number of data points in the sample. The standard deviation is calculated using the following formula.

$$\sqrt{s_{N-1}^2}$$

The following is the variance analysis for the selected stocks:

Ameris Bancorp (ABCB)

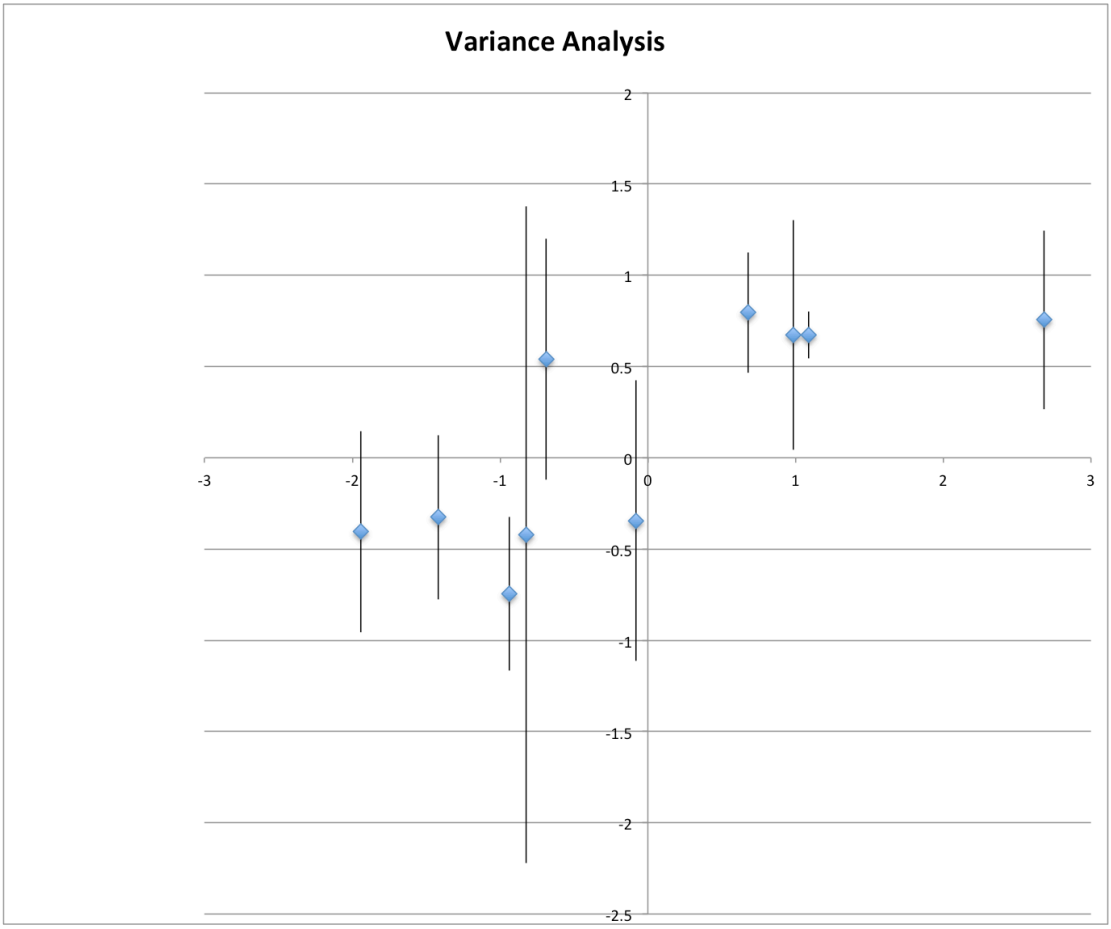


Figure 41 – Variance Analysis for Ameris Bancorp

Federal-Mogul Corp. (FDML)

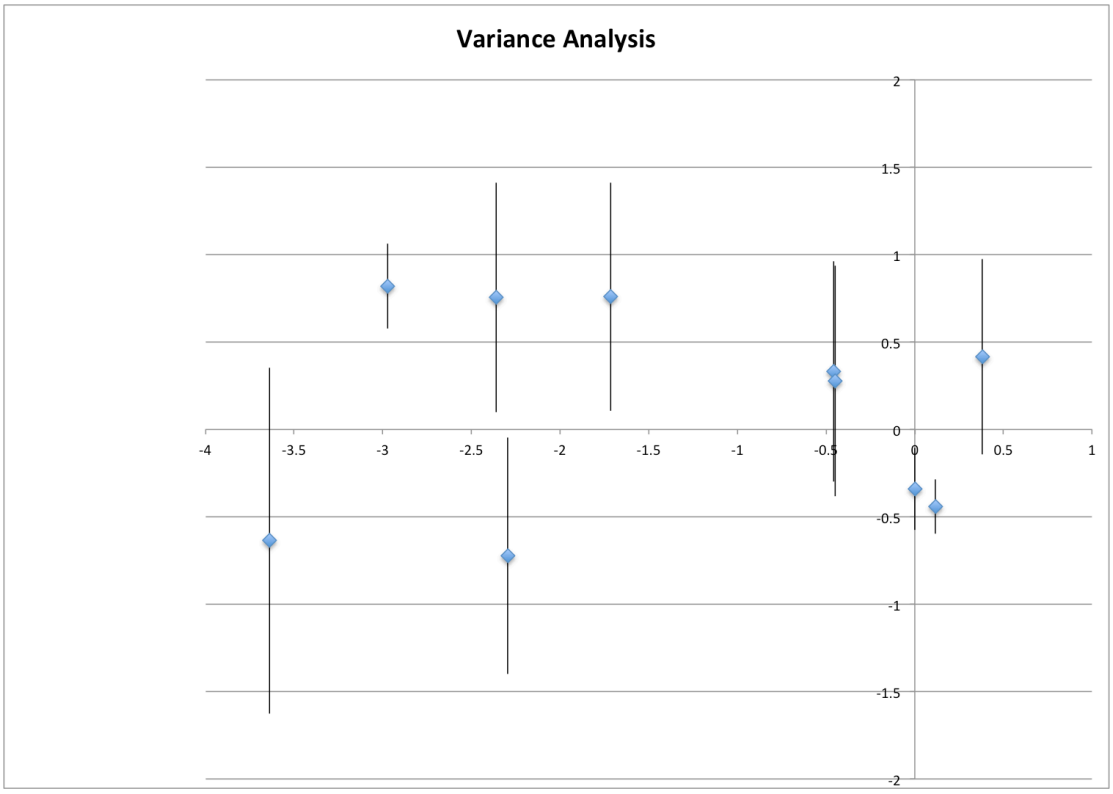


Figure 42 – Variance Analysis for Federal-Mogul Corp.

Microsoft Corporation (MSFT)

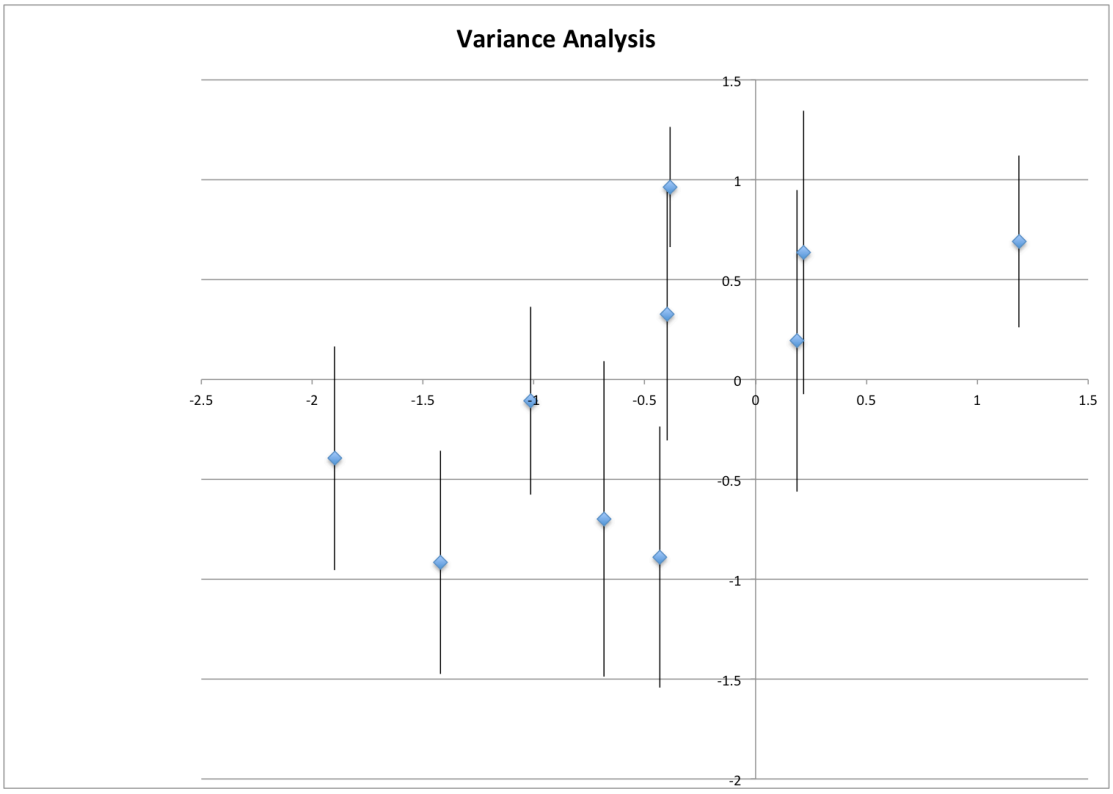


Figure 43 – Variance Analysis for Microsoft Corporation

Apple Inc. (AAPL)

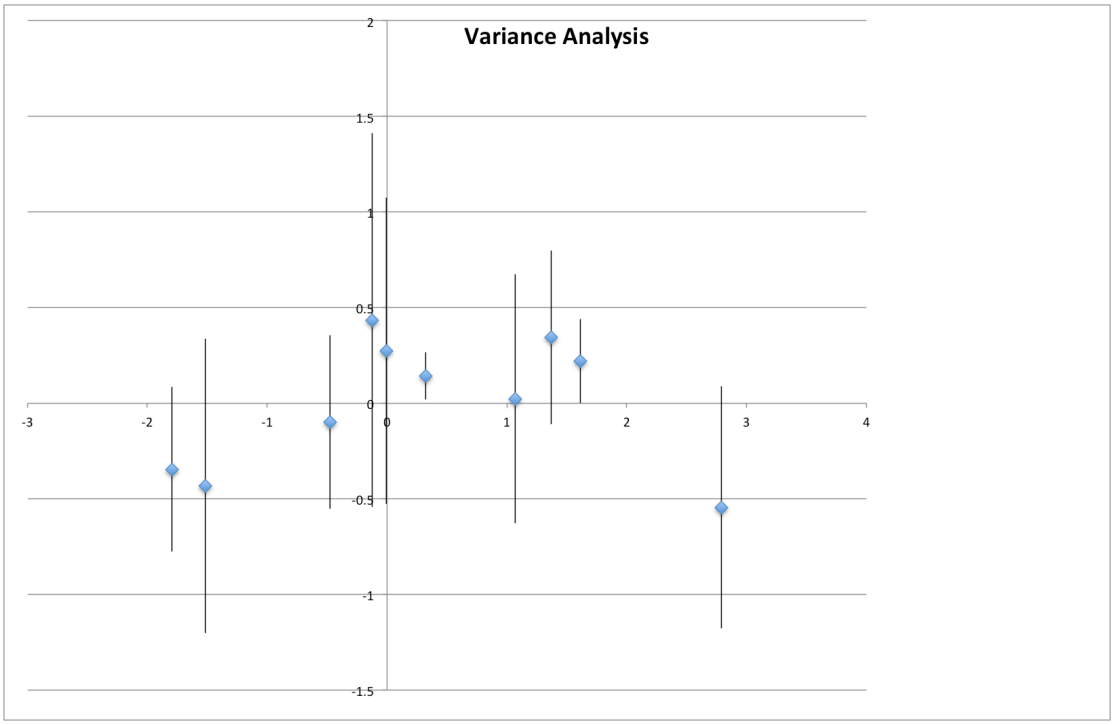


Figure 44 – Variance Analysis for Apple Inc.

5.4 Sentiment Analysis

A survey (see Appendix G) was sent out to selected people to help evaluate the effectiveness of the sentiment analysis. Each entry in the survey contains two options, upward or downward sentiment. The same articles were passed to the Sentiment Analysis component and the results were compared.

For each article, the opinions were aggregated and the majority opinion was noted.

For article 1 in the survey, the majority opinion was negative. The opinion from the system matched the survey result.

For article 2 in the survey, the majority opinion was positive. The opinion from the system matched the survey result.

For article 3 in the survey, the majority opinion was negative. The opinion from the system matched the survey result.

For article 4 in the survey, the majority opinion was positive. The opinion from the system matched the survey result.

For article 5 in the survey, the majority opinion was positive. The opinion from the system matched the survey result.

5.5 Website Evaluation

For evaluating the user experience, we launched the system in its beta stage to selected people. Comments and suggestions were gathered and the website was improved accordingly. A few users wanted a personal page where they can see their selected stocks and their actual and predicted trends on the same page. This feature was implemented and is a part of the user profile, called My Stocks.

Another feature requested was a way to communicate with other users. For this purpose a system wide chat was implemented which can be used by any user to talk about their predictions and subsequent discussions.

Another suggestion was to have a way to rate the users and their predictions. For this purpose, a user can rate another user as a reliable or unreliable source. In addition, a user can browse other user profiles based on the number of followers or the total rating.

The users also mentioned that they like the flow and smoothness of the website. Moreover, they liked the interaction of social networking and stock prediction and called it the next generation of stock prediction websites.

6. Discussion

The objective of our project was to implement an alternate way to predict stocks in response to the declining accuracy of the quantitative models. The system allows the users to analyze the prediction information in two manners. Firstly, the users can see the system-generated predictions for the different stocks. Secondly, the users can interact with the other users, receive investment advice and learn their investment strategies.

System testing along with the evaluation showed that the system has achieved nearly all the functionalities we planned to implement originally. In addition, some extra features have been implemented. The results show that users from Hong Kong with a university education had no difficulty using the system. However, due to time and financial limitations, we did not recruit testers from other backgrounds or with lower education levels to conduct a user acceptance test on a larger scale.

The evaluation shows that the system is a reliable system with acceptable average accuracy of 0.74. Moreover, correlation analysis between the actual and predicted trend shows an average correlation of 0.55 between the actual and predicted trend.

The following are some important points about the system:

- From the evaluation of the system, it can be inferred that stocks, which are well known and famous, have less accuracy and lower correlation between the actual and predicted trends compared to the lesser-known ones. Noise can be a possible reason for this. For example, for famous companies like Apple Inc., the sentiment analysis accuracy could be lower since people can talk about the company in many contexts. In contrast, for lesser-known companies, the discussion on the Internet is generally in the context of stock markets and therefore sentiment analysis could have a better accuracy.
- Initially, the focus of the system was both stocks and commodities. However, after analyzing the accuracy and correlation of commodities like Gold and Silver, we decided to leave out the commodities. This lower accuracy and correlation could be because Gold and Silver are common terms and therefore are less often mentioned in the context of stock markets on social networking sites like Twitter.

- The correlation can be an indication of what investment approach to adopt given the information from the system predictions. A positive correlation would imply that the actual stock price trend correlates to the system prediction, meaning one should invest according to the system prediction. A negative correlation could mean that one should adopt an investment approach that is the opposite of the system prediction. This negative correlation can be attributed to the manipulation of markets by highly influential traders and market makers. In other words, these influential people can influence investors, thereby moving the markets in a manner that would be advantageous to their investment positions.
- Since the data is gathered before the markets are open, the news and events during the market hours are not taken into account. For example, a company's quarterly earnings might be announced during the market hours and this can lead to a higher or lower share price. To tackle this problem, the system can also be run again during the market hours, giving new stock predictions. However, this process is very time and resource intensive meaning that the information gained from running the system may no longer be as timely. Given more system resources and finances, the system can be modified to run on multiple servers, allowing more real time predictions.
- For the sentiment analysis component, the data mining approach, classification could have been used. However, to achieve high accuracy for sentiment analysis, a huge training set is a prerequisite. Such a training set would consist of class labels and the words associated with them. Populating such a dataset is difficult using manual techniques. In addition, automatic techniques would have introduced a lot of noise and it would have led to lower accuracy. Due to the mentioned reasons we decided to use our current approach.

7. Conclusion

In this project, we designed and implemented a system that predicts stock trends using investors' expectations. Instead of using tools to forecast prices given market data, we have developed a tool that utilizes text from various investment websites as a metric for measuring consumer expectations for a particular stock. The system incorporates aspects of social networking to allow collaborative improvement of the system as well as user interaction.

There is potential for future developments that would enhance the system and increase its business value. The following items are some suggestions:

Relation Analysis

The link extractor component currently extracts all links from the specified sources. However, these links may not be relevant to the market we are focusing on. Moreover, some of the links from the source websites may not be related to the stock markets at all. Therefore, a dynamic downloading scheme is desired in the future works. This may involve a spider to search for information on the web, and some other component to identify whether the obtained article is related to the market we are observing. Some methods can be used to remove the unrelated articles. This would definitely improve the performance and accuracy of the system.

Quality of source list and word lists

The sources can heavily influence the predicted trend from the system. Therefore, it is important to choose the sources wisely. A component can be implemented to calculate the past accuracy of the sources in the system. The sources with higher past accuracy can be given a higher weighting in the sentiment analysis whereas sources with lower accuracy can be given a lower weighting.

The words in the upward and downward sentiment lists can also influence the accuracy of the prediction. Therefore, it is important to continue to improve these lists and include words that are used by investors to discuss stocks.

Other Properties

In this project, we focused only on the prediction of stock trends in the NASDAQ index. However, the financial news articles are influencing all kinds of indices. More index values maybe observed in the future to study the different influences that financial news articles had to the different market values. Besides predicting the index values, other stock properties such as volume, volatility, etc. can also be predicted.

Word Stemming

In the system, the irrelevant words are removed using the POS tagger program before sentiment analysis, but no stemming is done. Therefore, our positive sentiment, negative sentiment and negation word lists contain several different forms of the words, such as rose, rise, rises, risen etc. Stemming after POS would reduce the words to their root word, allowing us to have a shorter positive sentiment, negative sentiment and negation word lists, thereby improving performance.

Trust Value Calculation

A TrustValue Calculation [10] component can be implemented, which will allow the system to calculate how reliable and accurate a web source, or Twitter user is. This can be accomplished by keeping track of past predictions for a particular source and comparing it to actual market trends.

Trading Simulation

As the system is designed to be used by the users to make trading calls, it should be tested by doing simulation trading [11]. The return on such investments would be an important measurement of the value of the system.

Combination of technical analysis and sentiment analysis

One of the most popular methods for helping predict a stock's price is called Technical Analysis [12]. This method involves looking for patterns or indicators in stock prices, volumes, moving averages, etc. over time. It is hard for anyone to predict the future of the stock but this method can be effective in many cases because human beings are somewhat predictable. For example, when people see a stock start falling dramatically they often panic and sell their positions without investigating what

caused the fall. This causes even more people to sell their shares and this often leads to an "overshoot" of the stock price.

Another common technique involves Moving Averages. Many traders like to chart the 50-day and 200-day Moving Averages of their stock prices along with the prices themselves. When they see the current price cross over one of these Moving Averages on the charts it can be an indicator of a change in a long-term trend and it may be time to buy (or sell) the stock.

The combination of the above mentioned techniques along with the prediction from the system could serve as a more accurate predictor of stock markets since it would involve both technical analysis and investors' sentiments analysis.

Twitter Sentiment Analysis Improvements

In the system, there is no interface for the users to view the sentiment from a particular tweet processed by the system. However, for clarity and evaluation of the system, an interface can be implemented which will allow the users to view the sentiment for the individual tweets processed by the system. [13]

Twitter sentiment analysis can be made more accurate by the using the sentiments from the emoticons. In addition, to reduce redundancy of tweets, a component can be developed which removes the re-tweets. [14]

8. References

[1] Harvey, Daniel K. "*Forecasting the Belief of the Population: Prediction Markets, Social Media & Swine Flu.*"

Web. Accessed 21 Sept. 2011.

<http://homepages.inf.ed.ac.uk/miles/msc-projects/harvey.pdf>

[2] Twitter Inc.

Web. Accessed 11 Sept. 2011.

<http://twitter.com>

[3] Ma Yao, "*Financial Market Predictions Using Web Mining Approaches.*"

Web. Accessed 20 Sept. 2011.

<http://www.cse.ust.hk/~rossiter/ma_yao_mphil_thesis.pdf>

[4] Ailun Yi, "*Modeling the Stock Market Using Twitter.*"

Web. Accessed 31 Sept. 2011.

<http://homepages.inf.ed.ac.uk/miles/msc-projects/yi.pdf>

[5] Liang, Xun, and Rong-Chang Chen. "*Mining Stock News in CyberWorld Based on Natural Language Processing and Neural Networks.*"

Web. Accessed 19 Oct. 2011.

<http://www.cse.ust.hk/~leichen/courses/comp630p/collection/reference-5-13.pdf>

[6] Wikipedia. "*tf*idf*"

Web. Accessed 30 Nov. 2011.

<http://en.wikipedia.org/wiki/Tf*idf>

[7] Kurose, James F., and Keith W. Ross. "*Computer Networking: A Top-down Approach".* Upper Saddle River, NJ: Pearson Education, 2009. Print.

[8] Day, Alastair L. "*Mastering Financial Mathematics in Microsoft Excel."* Glasgow: Pearson Education Limited, 2005. Print.

[9] Wikipedia. "*Precision and Recall"*

Web. Accessed 31 Jan. 2012.

<http://en.wikipedia.org/wiki/Precision_and_recall/>

[10] Vivek Sehgal and Charles Song, "*SOPS: Stock Prediction using Web Sentiment."*

Web. Accessed 27 Dec. 2011.

<http://www.cs.umd.edu/~csfalcon/StockPrediction.pdf>

[11] Wolfram, Sebastian. "*Modelling the Stock Market Using Twitter."*

Web. Accessed 19 Feb. 2012.

<http://homepages.inf.ed.ac.uk/miles/msc-projects/wolfram.pdf>.

[12] HowTheMarketWorks.com. *"Stock Price Factors*."

Web. Accessed 09 April 2012.

<http://www.howthemarketworks.com/popular-topics/stock-price-factors.php>

[13] Go, Alec, Richa Bhayani, and Lei Huang. "*Twitter Sentiment."*

Web. Accessed 01 Apr. 2012.

<http://twittersentiment.appspot.com/>

[14] Go, Alec, Richa Bhayani, and Lei Huang. "*Twitter Sentiment Classification Using Distant Supervision*."

Web. Accessed 01 Apr. 2012.

<http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf>

Appendix A

Minutes of the 1st Group Meeting

Date: 7th June 2011

Time: 1:00 pm

Place: Library, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

Since this is the first meeting, there were no minutes of a previous meeting.

2. Report on Progress

Since this is the first meeting, there is no progress to be reported.

3. Discussion Items

- Scope of the project
- Programming languages for implementation of spider & HTML Extractor
- Website interface and how prediction results will be displayed on the website

4. Meeting Adjournment

The meeting was adjourned at 3:00 p.m.

Minutes of the 2nd Group Meeting

Date: 17th August 2011

Time: 2:30 pm

Place: Café, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Nikhil has started implementing the spider.

Mothusi has started learning the Python and Java programming languages.

3. Discussion Items

- Brainstormed the required components for the system
- Outlined algorithm for the spider
- Discussed the possible utility of Python for text analysis

4. Meeting Adjournment

The meeting was adjourned at 5:00 p.m.

Minutes of the 3rd Group Meeting

Date: 7th September 2011

Time: 1:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Nikhil has started working on the algorithm for extracting text from a webpage.

Mothusi has started collecting information about different financial resources.

3. Discussion Items

- Detailed discussion about the different components of the system
- Discussed possible algorithms that can be used for our text analysis.

4. Meeting Adjournment

The meeting was adjourned at 2:00 p.m.

Minutes of the 4th Group Meeting

Date: 19th September 2011

Time: 3:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Nikhil has almost completed implementing the link extractor and text extractor. Testing will begin shortly.

Mothusi has started learning about the Twitter API.

3. Discussion Items

- An in depth discussion was carried out with Professor Rossiter regarding how the spider and text extractor should work together as a system and the sources we could use for gathering financial information.
- Nikhil mentioned about the algorithm for using the Twitter API.

4. Meeting Adjournment

The meeting was adjourned at 4:00 p.m.

Minutes of the 5th Group Meeting

Date: 6th January 2012

Time: 3:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Professor David Rossiter

Absent: Mothusi Majinda

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Nikhil and Mothusi are currently working on completing the text analysis component and using JDBC to send prediction values to the database.

3. Discussion Items

- Nikhil met Professor Rossiter to discuss the progress of our implementation.
- Nikhil sought advice for the text analysis component
- Set date for first system demo in the first week of February.

4. Meeting Adjournment

The meeting was adjourned at 4:00 p.m.

Minutes of the 6th Group Meeting

Date: 8th February 2012

Time: 1:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Professor David Rossiter

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

We have completed the implementation of the server side Java code. We are currently working on the front end website interface.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Received feedback on how to improve. This includes:
  - System administrator features
  - User features
  - Using Google search to get better sources for link extractor
  - Add depth parameter for link extractor
  - Compare system performance with actual market performance

4. Meeting Adjournment

The meeting was adjourned at 1:45 p.m.

Minutes of the 7th Group Meeting

Date: 24th February 2012

Time: 1:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

We have completed the implementation of the layout for the website component. Work on the backend of the website has started.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Received feedback on how to improve. This includes:
  - Social element
  - User interaction
  - User Collaboration
  - Compare system performance with actual market performance

4. Meeting Adjournment

The meeting was adjourned at 2:00 p.m.

Minutes of the 8th Group Meeting

Date: 28th February 2012

Time: 3:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The implementation of the backend of the website is half completed.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Discussed how to evaluate the system. This included various statistical possibilities, beta for user feedback on the website interface and a survey to compare system sentiment analysis to human sentiment analysis.

4. Meeting Adjournment

The meeting was adjourned at 4:00 p.m.

Minutes of the 9th Group Meeting

Date: 8th March 2012

Time: 2:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The implementation of the user profile and administrative controls has started.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Designed the survey for sentiment analysis evaluation. This involved gathering articles online.

4. Meeting Adjournment

The meeting was adjourned at 3:00 p.m.

Minutes of the 10th Group Meeting

Date: 13th March 2012

Time: 1:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The implementation of the user profile and administrative controls has been completed. Implementation of the chat feature has begun.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Discussed about the different administrator levels and banning/unbanning the users.
- Discussed about the Prices and Variance Component.

4. Meeting Adjournment

The meeting was adjourned at 1:45 p.m.

Minutes of the 11th Group Meeting

Date: 18th March 2012

Time: 12:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Implementation of the chat feature has been completed. Implementation of the different administrator levels has also been completed.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Discussed how the graphical interpretation can be made more interactive and how the reliability of the system can be improved.

4. Meeting Adjournment

The meeting was adjourned at 1:00 p.m.

Minutes of the 12th Group Meeting

Date: 23rd March 2012

Time: 3:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The graphical representation of the predicted trend has been made more interactive. Implementation of the prices and variance component has been completed.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Discussed how the users can browse through other users' profiles and choose to follow them.

4. Meeting Adjournment

The meeting was adjourned at 3:45 p.m.

Minutes of the 13th Group Meeting

Date: 28th March 2012

Time: 12:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor David Rossiter

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The follow feature and user rating feature has been implemented.

3. Discussion Items

- Showed Professor Rossiter a demo of our system.
- Discussed about the final report and the evaluation of the system.
- We conducted the survey we designed for evaluating system analysis. We ran the articles through the system sentiment analysis and compared it to the survey results.

4. Meeting Adjournment

The meeting was adjourned at 12:45 p.m.

Minutes of the 14th Group Meeting

Date: 1st April 2012

Time: 1:00 pm

Place: Cafe, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

The work on the final report has started.

3. Discussion Items

- Discussed about the final report and its structure.
- Discussed the shortcomings of the system and how they can be improved in the future work. This includes Twitter improvements, trading simulation and trust value calculations, etc.

4. Meeting Adjournment

The meeting was adjourned at 2:45 p.m.

Minutes of the 15th Group Meeting

Date: 7th April 2012

Time: 5:00 pm

Place: Library, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Progress has been made on the final report.

3. Discussion Items

- Discussed further about the final report and its structure.
- Discussed about how the email component can be implemented. Chose to implement this on the Java side rather than PHP since emails would be sent immediately after the system has completed running.

4. Meeting Adjournment

The meeting was adjourned at 6:45 p.m.

Minutes of the 16th Group Meeting

Date: 11th April 2012

Time: 1:00 pm

Place: Professor Rossiter's Office, HKUST

Attending: Nikhil Berry, Mothusi Majinda, Professor Rossiter

Absent: None

Recorder: Nikhil Berry

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Progress has been made on the final report. The email component has been implemented.

3. Discussion Items

- Discussed further about the final report and its structure.
- Showed Professor Rossiter a demo of the website.
- Discussed how the stocks can be color coded in the email. Developed a schema for the same.

4. Meeting Adjournment

The meeting was adjourned at 2:00 p.m.

Minutes of the 17th Group Meeting

Date: 13th April 2012

Time: 1:00 pm

Place: Library, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Progress has been made on the final report.

3. Discussion Items

- Worked on the Appendix section of the Final Report.
- Had difficulty generating the Java documentation. We found an Eclipse plugin to automatically generate the documentation. Documentation is only exported in HTML. Tried to find way to convert HTML to .doc or rtf. However, some content in the generated Java documentation made this impossible to do automatically. Eventually had to copy paste the documentation from the HTML generated files into the report and manually format it.

4. Meeting Adjournment

The meeting was adjourned at 5:00 p.m.

Minutes of the 18th Group Meeting

Date: 17th April 2012

Time: 1:00 pm

Place: Library, HKUST

Attending: Nikhil Berry, Mothusi Majinda

Absent: None

Recorder: Mothusi Majinda

1. Approval of Minutes

The minutes of the last meeting were approved without amendment.

2. Report on Progress

Progress has been made on the final report.

3. Discussion Items

- Worked in the Evaluation section of the Final Report.
- Had difficulty plotting varying error bars for different points in excel graphs. Research was conducted to resolve the issue. Solution found by creating custom data series to hold the error bar values.

4. Meeting Adjournment

The meeting was adjourned at 7:00 p.m.

Appendix B

PROJECT PLANNING

1. Division of Work

Task	Nikhil Berry	Mothusi Majinda
Requirements Analysis	L	A
Twitter API Functionality	L	A
Link Extractor	L	A
Text Extractor	L	A
Sentiment Analysis	L	A
Tagger	A	L
Database Design and Implementation	A	L
Website Layout	L	A
System integration in the website	L	A
Requests component of the website	L	A
Collaborate component of the website	A	L
Graphical Representation of trend in the website	A	L
Administrative Controls in the website	A	L
Proposal Report	A	L
Progress Report	L	A
System Testing and Debugging	A	L
Evaluation	A	L
Final Report and Poster	L	A
Project Presentation	A	L

L: Leader **A**: Assistant

Table 7: Division of work

2. Gantt Chart

Below is the Gantt chart of our project schedule.

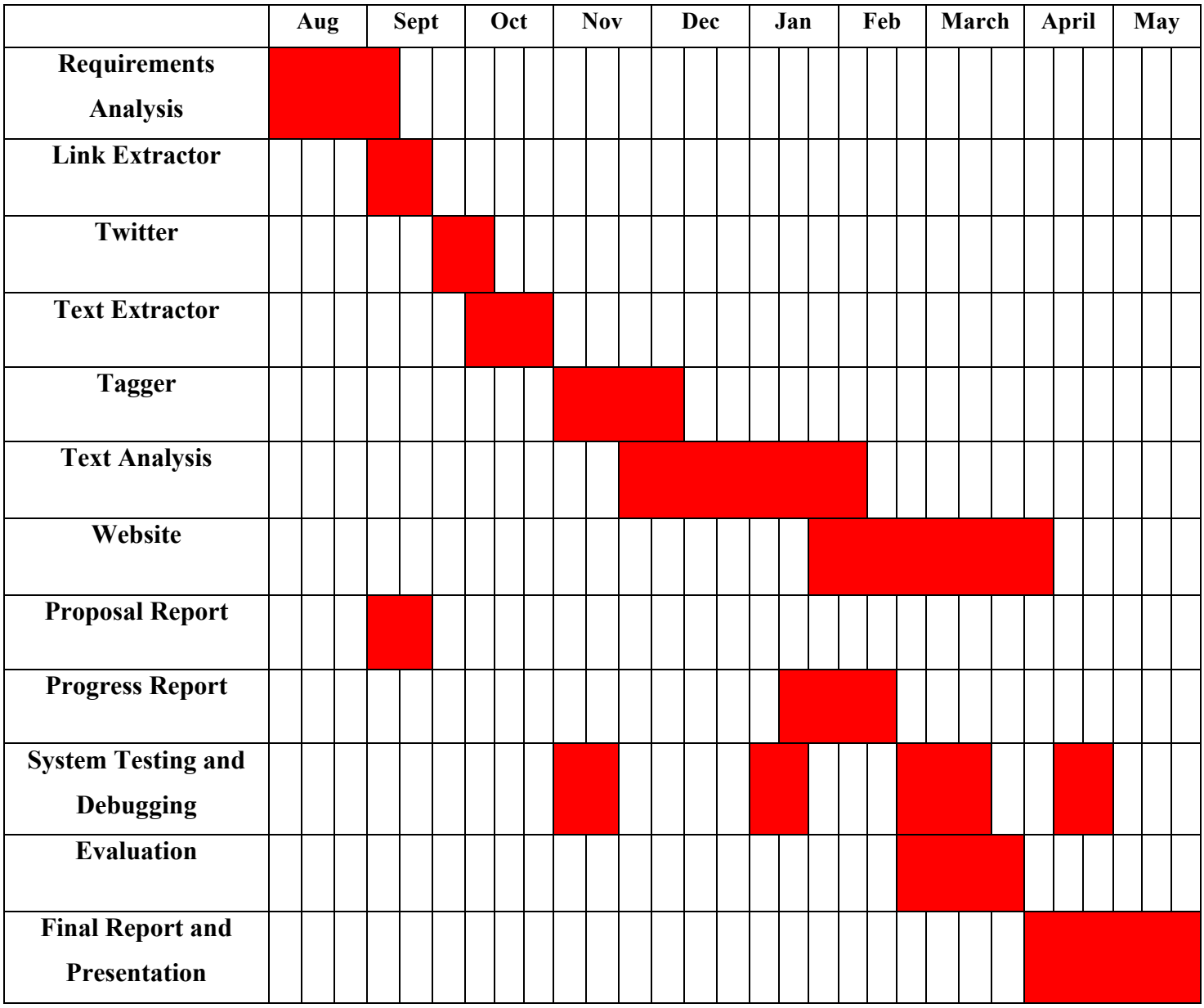


Table 8: Gantt Chart

Appendix C

The following are the lists of the hardware and software requirements to run the system.

Server-side Requirements

Server Hardware:

- Hard Drive: 1 TB 7200 RPM
- RAM: 8GB 1067 MHz DDR3
- Processor: 2.4GHz quad-core Intel Core i7
- Bandwidth: 1 TB per month

Server Software Support:

- Operating System: Linux (CentOS)
- Web Server: Apache
- Database: MySQL
- Programming Languages: Java, PHP

Client-side Requirements

Client Hardware:

- Hard Drive: 250GB
- RAM: 1GB 1067 MHz DDR3
- Processor: Intel Core 2 Duo

Client Software Support:

- Operating System: Any
- Web Browser: Any with JavaScript support

Appendix D: Domain Model

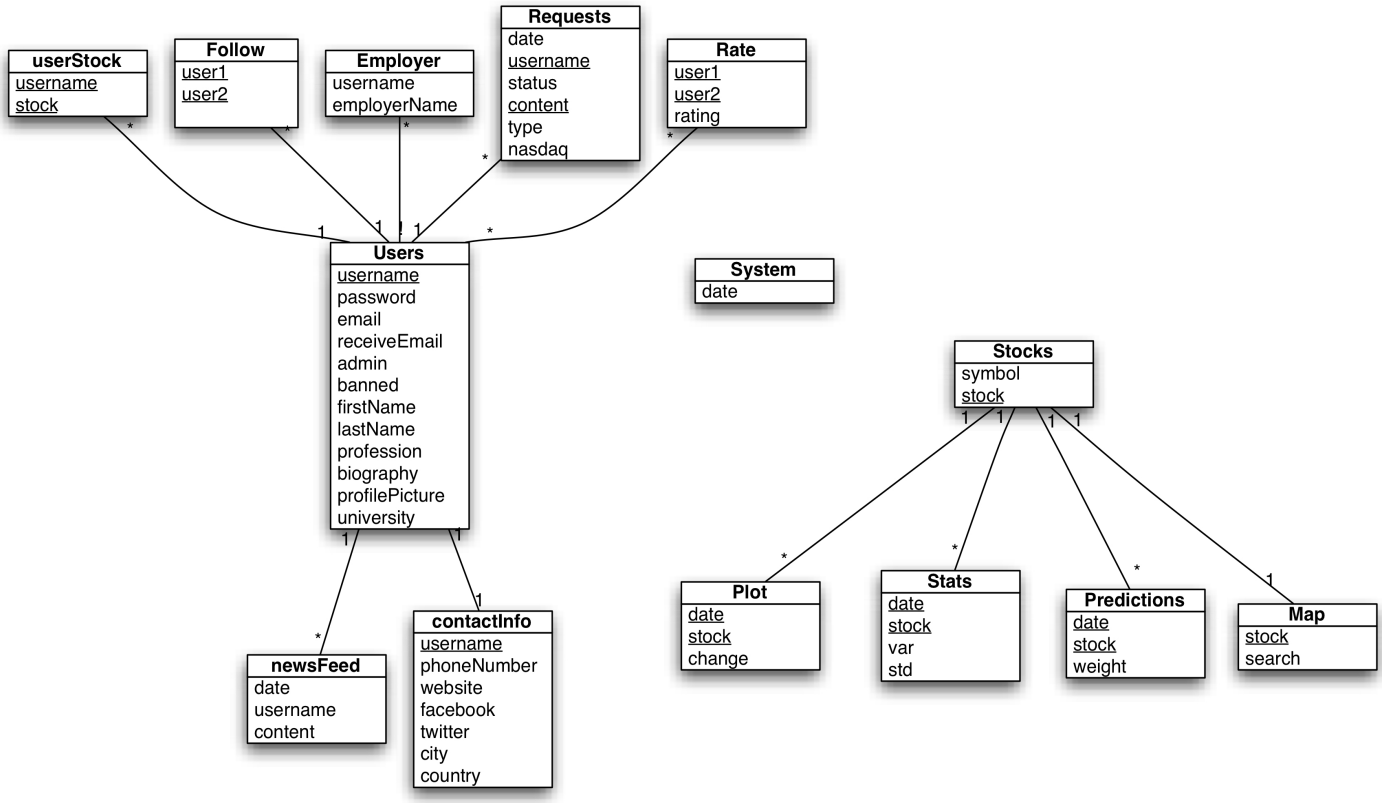
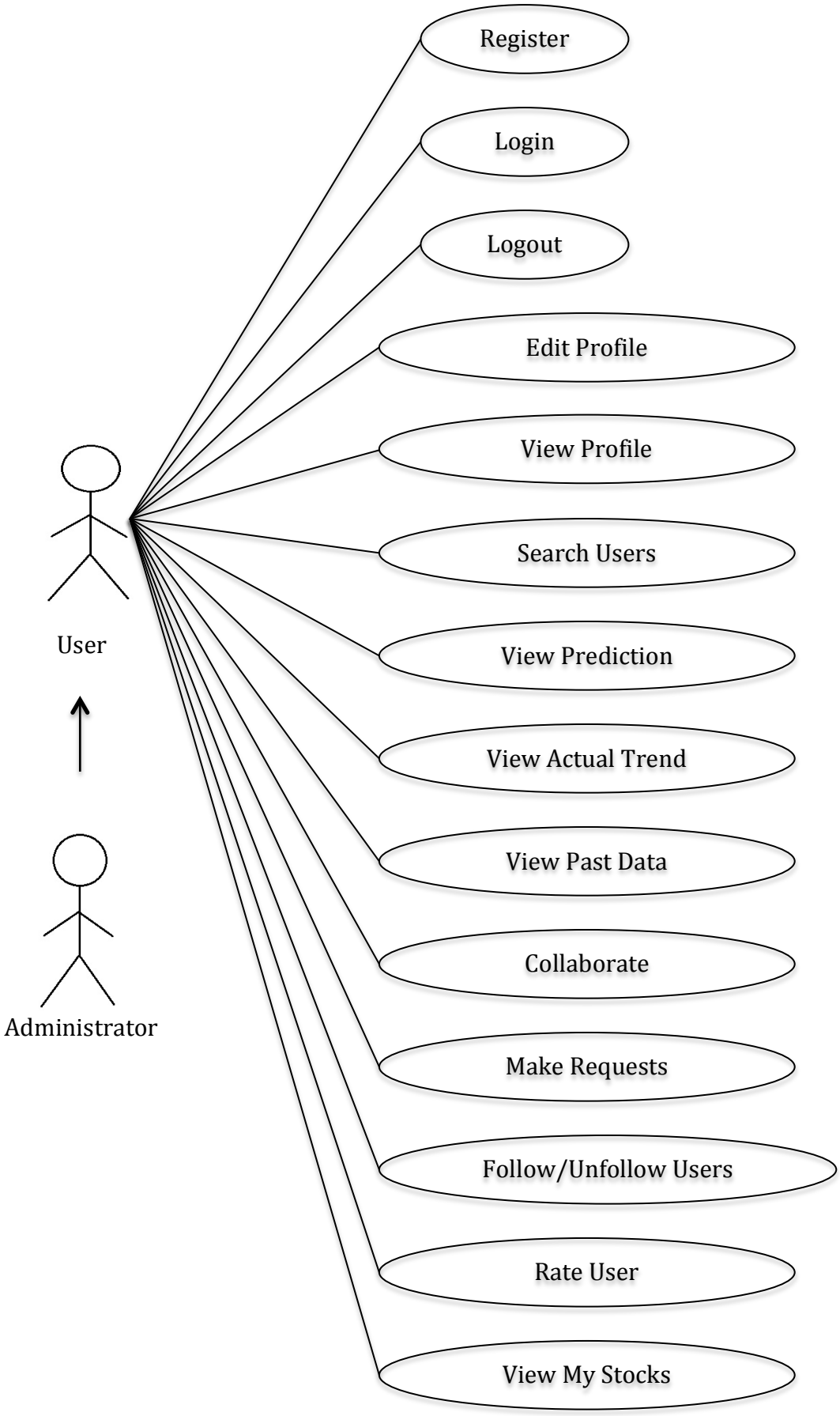
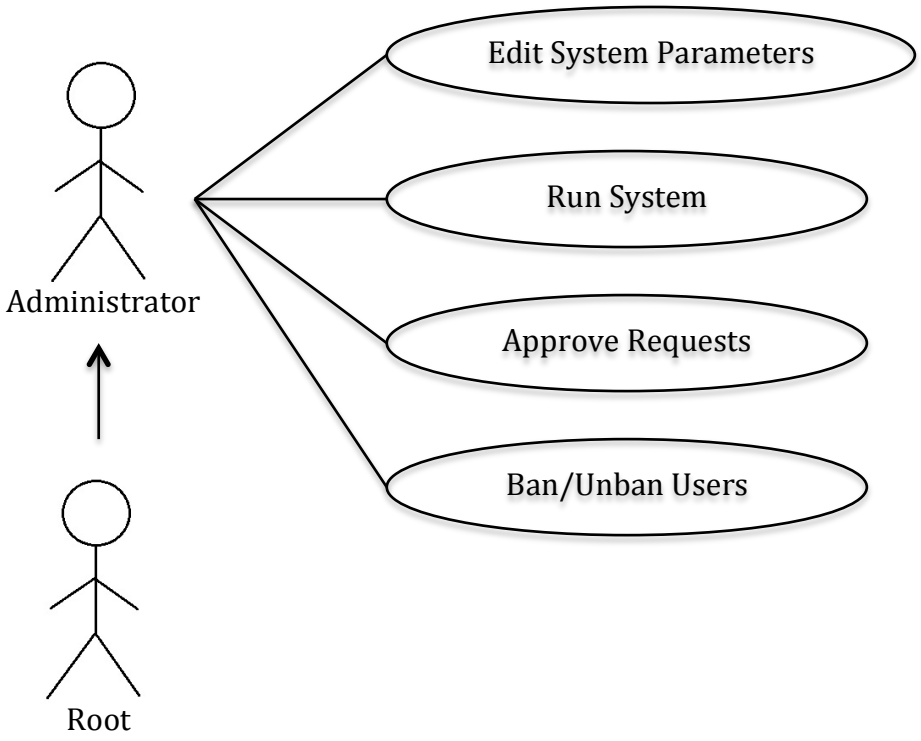


Figure 45 – Domain model

Appendix E: Use Case Model




Edit System Parameters
Run System
Approve Requests
Ban/Unban Users
Administrator
Root

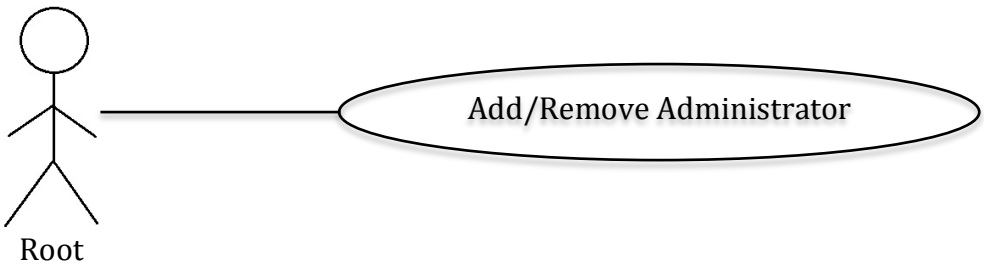


Figure 46 – Use Case Model

Use Case	*Login*
Description	*The user enters their username and password to login.*
Actors	*User, Administrator, Root*
Assumptions	*Login session is inactive.*

Use Case	*Logout*
Description	*The user enters their username and password to login.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Register*
Description	*The user completes the registration fields.*
Actors	*User*
Assumptions	*The user does not currently have an account.*

Use Case	*Edit Profile*
Description	*The user can edit their profile information.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*View Profile*
Description	*The user can view their profile information.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Search Users*
Description	*The user can search for other users. The search can be a search by number of followers or total rating.*
Actors	*User, Administrator, Root*
Assumptions	*Login session is inactive.*

Use Case	*View Prediction*
Description	*The user can view the system predictions for different stocks.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*View Actual Trend*
Description	*The user can view the actual stock price trends for different stocks.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Collaborate*
Description	*The user can view words generated by the POS and suggest them as positive or negative sentiment words.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Make Request*
Description	*A user can make a request to add new sources, negation words, negative sentiment words and positive sentiment words. If the user is an administrator or root, the request is automatically approved.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Rate User*
Description	*A user can rate another user as reliable or unreliable.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*View My Stocks*
Description	*A user can have a list of stocks they follow called my stocks. On this page, they can view the actual prices of the stocks as well as the most recent system prediction for the stocks.*
Actors	*User, Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Edit System Parameters*
Description	*The user can edit the parameters used for sentiment analysis (positive sentiment words, negative sentiment words, sources and negation words).*
Actors	*Administrator, Root*

Assumptions	*The user is logged in.*

Use Case	*Run System*
Description	*The user can run the sentiment analysis system. They can view the files as they are generated real time and also view a log of all past generated files.*
Actors	*Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Approve Requests*
Description	*The user can approve or requests made by regular users. The user can also view past approved or rejected requests.*
Actors	*Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Ban/Unban User*
Description	*The user can ban or unban regular users of the system. If the user root, they can also ban administrator users.*
Actors	*Administrator, Root*
Assumptions	*The user is logged in.*

Use Case	*Make/Remove Administrator*
Description	*The user can add or remove administrators.*
Actors	*Root*
Assumptions	*The user is logged in.*

Appendix F: Java Code Documentation

Class Driver

java.lang.Object
└ **Driver**

public class **Driver**
extends java.lang.Object
The Class Driver.

Author:
Mothusi

Nested Class Summary	
class	**Driver.Extension** The Class Extension.

Field Summary	
boolean	**thr** The thread variable.

Constructor Summary
Driver()

Method Summary	
void	**delete**(java.io.File file) Delete.
void	**file**() File.
void	**initialize**() Initialize.
static void	**main**(java.lang.String[] args) The main method.
void	**twitter**() Twitter.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

thr

public boolean **thr**
Thread variable.

Constructor Detail

Driver

public **Driver**()

Method Detail

main

public static void **main**(java.lang.String[] args)
The main method.

Parameters:
args - the arguments

twitter

public void **twitter**()
Twitter.

initialize

public void **initialize**()
Initialize.

delete

public void **delete**(java.io.File file)
throws java.io.IOException
Delete.

Parameters:
file - the file
Throws:
java.io.IOException - Signals that an I/O exception has occurred.

file

public void **file**()
File.

Class Database

java.lang.Object
└ **Database**

public class **Database**
extends java.lang.Object
The Class Database.

Constructor Summary
Database() Instantiates a new database.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Database

public **Database**()
throws java.lang.Exception
Instantiates a new database.

Throws:
java.lang.Exception - the exception

Method Detail

main

public static void **main**(java.lang.String[] args)

The main method.

Parameters:
args - the arguments

Class email

java.lang.Object
└ **email**

public class **email**
extends java.lang.Object
The Class email.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

main

public static void **main**(java.lang.String[] args)

The main method.

Parameters:
args - the arguments

Class Google

java.lang.Object
└ **Google**

public class **Google**
extends java.lang.Object
The Class Google.

Constructor Summary
Google() Instantiates a new google.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Google

public **Google**()
Instantiates a new google.

Method Detail

main

public static void **main**(java.lang.String[] args)
The main method.

Parameters:
args - the arguments

Class googleExtract

java.lang.Object
└ **googleExtract**

public class **googleExtract**
extends java.lang.Object
The Class googleExtract.

Constructor Summary
googleExtract()

Method Summary	
void	**extract**(java.lang.String x) Extract.
void	**link**() Link.
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

googleExtract

public **googleExtract**()

Method Detail

extract

public void **extract**(java.lang.String x)
throws java.lang.Exception
Extract.

Parameters:
x - the x
Throws:
java.lang.Exception - the exception

link

public void **link**()
throws java.lang.Exception
Link.

Throws:
java.lang.Exception - the exception

main

public static void **main**(java.lang.String[] args)
The main method.

Parameters:
args - the arguments

Class linkextract

java.lang.Object
└ **linkextract**

public class **linkextract**
extends java.lang.Object
The Class linkextract.

Constructor Summary
linkextract()

Method Summary	
void	**extract**(java.lang.String x) Extract.
void	**link**() Link.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

linkextract

public **linkextract**()

Method Detail

extract

public void **extract**(java.lang.String x)
throws java.lang.Exception
Extract.

Parameters:
x - the x
Throws:
java.lang.Exception - the exception

link

public void **link**()
throws java.lang.Exception
Link.
Throws:
java.lang.Exception - the exception

Class Prices

java.lang.Object
└ **Prices**

public class **Prices**
extends java.lang.Object
The Class Prices.

Constructor Summary
Prices() Instantiates a new prices.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.
static void	**text**(java.lang.String source, java.lang.String Stock) Text.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Prices

public **Prices**()
Instantiates a new prices.

Method Detail

text

public static void **text**(java.lang.String source,
java.lang.String Stock)
Text.

Parameters:
source - the source
Stock - the stock

main

public static void **main**(java.lang.String[] args)
The main method.

Parameters:
args - the arguments

Class tagger

java.lang.Object
└ **tagger**

public class **tagger**
extends java.lang.Object
The Class tagger.

Constructor Summary
tagger(java.lang.String File, edu.stanford.nlp.tagger.maxent.MaxentTagger tagger) Instantiates a new tagger.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

tagger

public **tagger**(java.lang.String File,
edu.stanford.nlp.tagger.maxent.MaxentTagger tagger)
Instantiates a new tagger.

Parameters:
File - the file
tagger - the tagger

Method Detail

main

public static void **main**(java.lang.String[] args)

The main method.

Parameters:
args - the arguments

Class TextExtract

java.lang.Object
└ **TextExtract**

public class **TextExtract**
extends java.lang.Object
The Class TextExtract.

Constructor Summary
TextExtract()

Method Summary	
void	**text**(java.lang.String source) Text.
void	**textextract**() Textextract.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

TextExtract

public **TextExtract**()

Method Detail

text

public void **text**(java.lang.String source)
Text.

Parameters:
source - the source

textextract

public void **textextract**()
throws java.lang.Exception
Textextract.

Throws:
java.lang.Exception - the exception

Class Tree
java.lang.Object
└ **Tree**

public class **Tree**
extends java.lang.Object
The Class Tree.

Constructor Summary
Tree(java.lang.String inputFile) Instantiates a new tree.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Tree

public **Tree**(java.lang.String inputFile)
throws java.io.IOException
Instantiates a new tree.

Parameters:
inputFile - the input file
Throws:
java.io.IOException - Signals that an I/O exception has occurred.

Method Detail

main

public static void **main**(java.lang.String[] args)

The main method.

Parameters:
args - the arguments

Class twitter

java.lang.Object
  └ **twitter**

public class **twitter**
extends java.lang.Object
The Class twitter.

Constructor Summary
twitter(java.lang.String query) Instantiates a new twitter.

Method Summary

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

twitter

public **twitter**(java.lang.String query)
  throws java.lang.Exception
Instantiates a new twitter.

Parameters:
query - the query
Throws:
java.lang.Exception - the exception

Class Variance

java.lang.Object
└ **Variance**

public class **Variance**
extends java.lang.Object
The Class Variance.

Constructor Summary
Variance() Instantiates a new variance.

Method Summary	
static void	**main**(java.lang.String[] args) The main method.

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Variance

public **Variance**()
Instantiates a new variance.

Method Detail

main

public static void **main**(java.lang.String[] args)
The main method.

Parameters:
args - the arguments

Appendix G: Survey

Oil prices fell in early trade on reports that Iran and six major powers held "constructive" talks on Iran's nuclear program over the weekend, said IG Markets analyst Justin Harper.

Positive Sentiment Negative Sentiment

Positive growth forecast from the IMF will lead to upbeat sentiments in the global markets today and on account of this we expect the US dollar index to remain weak. Taking cues from this, we expect gold and silver prices to trade higher today. However, a stronger Rupee may cap gains on the domestic bourses.

Positive Sentiment Negative Sentiment

(Reuters) - Goldman Sachs Group Inc (NYS:GS - News) posted a 23 percent decline in quarterly earnings after it dialed down risk-taking in tricky markets, and clients also reduced their appetite for bets.

Positive Sentiment Negative Sentiment

TOKYO (Reuters) - Asian shares rose on Wednesday as firm demand at Spanish debt sales and positive U.S. corporate earnings boosted investor confidence in riskier assets.

Positive Sentiment Negative Sentiment

Cheniere Energy (NYSE: LNG) traded 3.6% higher today on news that the federal government approved the company to construct a natural gas export facility in Louisiana. With nat gas prices in the U.S. trading at long-term lows, natural gas companies hope to capitalize on higher prices abroad, a move that would be a boon for companies operating in the industry.

Positive Sentiment Negative Sentiment

Appendix H: Positive Sentiment Word List

Acquisition	Dividends	Incrementation	Success
Acquisitions	Expand	Incrementations	Successful
Affirmative	Expanded	Incremented	Surge
Appreciate	Expanding	Incrementing	Surged
Appreciated	Expands	Increments	Surges
Appreciates	Expansion	Optimal	Surging
Appreciating	Flourish	Optimum	Thrive
Appreciation	Flourished	Perfect	Thrives
Assured	Flourishes	Positive	Thriving
Bargain	Flourishing	Positives	Up
Bargains	Gain	Profit	Upgrade
Boom	Gained	Profitable	Upgraded
Boomed	Gains	Profiting	Upgrades
Booming	Good	Profits	Upgrading
Booms	Grew	Prosper	Upsurge
Boost	Grow	Prospering	Upsurged
Booster	Growing	Prospers	Upsurges
Boosting	Grows	Purchase	
Boosts	Growth	Purchases	
Bull	Growths	Raise	
Bullish	Hedge	Raises	
Buy	Hedged	Rallies	
Buys	Hedging	Rally	
Climb	High	Real	
Climbed	Higher	Recoveries	
Climbs	Improve	Recovery	
Concrete	Improved	Rise	
Confidence	Improvement	Risen	
Confidences	Improves	Rises	

Table 9: Positive Sentiment word list

Appendix I: Negative Sentiment Word List

Adverse	Crack	Devaluated	Fell
Adverses	Crunch	Devaluating	Flop
Adversity	Debacle	Devaluation	Hardship
Atrocious	Debt	Disaster	Horrible
Bad	Debts	Disruption	Inferior
Bankrupt	Decline	Disruptions	Kleptocracy
Bankruptcy	Declined	Distress	Losing
Bankrupted	Declining	Disturbance	Losses
Bankrupting	Decrease	Down	Lost
Barrier	Default	Downfall	Mess
Barriers	Defaulted	Downgrade	Misadventure
Bear	Defaulting	Downgraded	Mischance
Bearish	Deficit	Downgrading	Misfortune
Below	Deficited	Downhill	Mishap
Bottomward	Deficits	Downturn	Negative
Burst	Deflation	Downturned	Recession
Bust	Deflations	Downturns	Recessions
Buster	Depreciate	Downward	Retreated
Busts	Depreciated	Dropped	Risk
Calamity	Depreciation	Dropping	Risks
Cascade	Depress	Dump	Ruin
Cascaded	Depressed	Dumped	Ruination
Cascading	Depression	Dumper	Ruined
Cataclysm	Descend	Dumping	Ruining
Cataclysmic	Descended	Failure	Sell
Catastrophe	Descending	Fall	Selling
Collapse	Devaluate	Falling	Sink

Table 10: Negative Sentiment word list

Sinking	Slump	Tragedy	Weakening
Slid	Slumped	Under	Woe
Slide	Slumping	Underneath	Worse
Slides	Stagnate	Undoing	Worsen
Sliding	Stagnation	Volatile	Worst
Slip	Subjacent	Volatilities	Wreck
Slipping	Terrible	Volatility	

Appendix J: Negation Word List

Aren't	Couldn't	Isnt	Wont
Arent	Couldnt	Never	Wouldn't
Can't	Don't	Not	
Cannot	Dont	Shouldn't	
Cant	Isn't	Won't	

Table 11: Negation word list

Appendix K: Source List

http://money.cnn.com/
http://www.thisismoney.co.uk/money/index.html
http://www.thestreet.com/
http://www.smartmoney.com/
http://seekingalpha.com/
http://www.bloomberg.com/
http://www.forbes.com/
http://finance.yahoo.com/
http://www.economist.com/
http://www.kiplinger.com/
http://www.nasdaq.com/
http://online.wsj.com/
http://www.ft.com/
http://www.efinancialnews.com/
http://www.reuters.com/
http:// www.cnbc.com/
http://www.foxbusiness.com/index.html
http://www.businessweek.com/
http://www.ibtimes.com/
http://www.guardian.co.uk/business
http://www.fool.com/
http://www.investopedia.com/
http://www.minyanville.com/
http://www.cbsnews.com/moneywatch/

Table 12: Source list