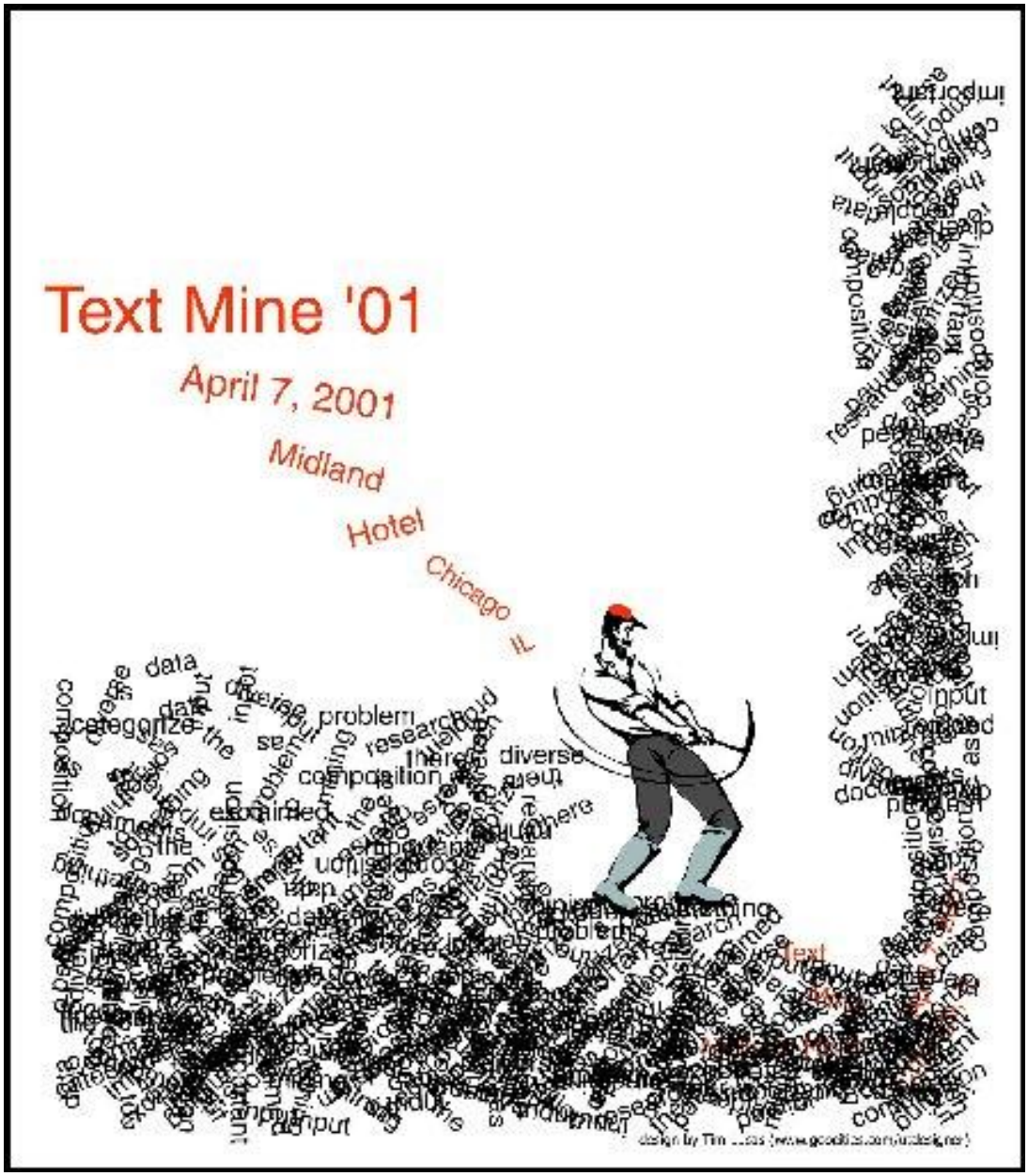
Text Mining Project



Ofir Marcus 201618469

Eliezer Steinbok 336181490

Danit Lev 301643441

Talya Sohlberg 021987631

Introduction

Text mining is one of fasted growing fields of computer science in the last decade. The field gives a whole new meaning to types of data that were neglected before and helps us find "hidden" patterns in raw data.

Sentiment analysis is a sub-topic of text mining. The purpose of sentiment analysis is to determine the attitude of the speaker concerning some topic or context. Sentiment analysis is used in various fields as a data analysis tool. As new media types emerge and new platforms become available for people to express their opinion, sentiment analysis steps in with automatic tools to analyze different types of written data.

Our project consists of a system that performs sentiment analysis on stock Tweets, based on data from StockTwits.com. StockTwits is based on the extremely popular Twitter which is a platform allowing people to share and express their opinion on events or topics. Hundreds of millions of Tweets are posted daily on Twitter with each Tweet containing under 140 characters. StockTwits.com is less popular than Twitter, but the site still has thousands of Tweets per day. These Tweets are enough to express a sharp opinion over a topic. Assuming there is a correlation between people's opinion (and Tweets) about a stock and that a Tweet may influence other people, StockTwits data may be extremely valuable for trading stocks.

Our system attempts to evaluate the sentiment of a stock based on Tweets and can be used to make predictions of future fluctuations of a stock price.

The data we used in the system includes two years of all the Tweets on StockTwits.com that mention a publicly traded US company.

The input to our system is a Tweet and the output is a sentiment for each ticker mentioned in the Tweet. The sentiment is one of the set: {very positive, positive, neutral, negative, very negative}.

Analysis method

We will now explain the technical details of the implementation of our sentiment analysis system.

One of the major challenges of analyzing Tweets is that people write in a very terse style of language. Since Tweets are limited to 140 characters, Tweets usually contain grammatical errors, spelling mistakes and slang. For this reason and others, analyzing Tweets with POS (Part Of Speech) taggers or semantic parsers is extremely difficult as most of them assume a perfectly written text.

Another challenge one encounters when attempting to analyze Tweets is that some Tweets contain multiple tickers and a different sentiment must be given to each ticker in the Tweet. This requires breaking the sentence up into its different parts. We solved this problem by declaring a list of "separators". These are characters and words which break up a sentence into multiple parts. A simple example of a separator is a single period (".”). We assume that each side of the period refers to a different ticker. If two tickers appear in the same part of a sentence we give each ticker the same sentiment. If after we have split the Tweet into multiple parts there are no tickers in a certain part of the Tweet, we ignore this split. We only split Tweets into multiple parts if the Tweet contains more than one ticker.

For example, the Tweet: "$AAPL's iPhone 5 sucks. I like Nexus 4 $GOOG much more" is split into two parts: "$AAPL's iPhone 5 sucks" and "I like Nexus 4 $GOOG much more". The separator in this example is a period and our system analyses each part separately, giving $AAPL a negative sentiment and $GOOG a positive one.

After we have split the Tweet up into its multiple parts, we replace all abbreviations with their unabbreviated form. For example, "OMG" is replaced with "Oh my God". "R" is replaced with "are". "U" is replaced with "you". The replacement is done using a dictionary called the SlangLookupTable. This dictionary simply maps abbreviated words to their full, unabbreviated form. This stage is very important for analyzing Tweets because Tweets are written in informal language and often contain abbreviations.

The final stage is to actually give a sentiment to each ticker in a Tweet. To do this we used a file called EmotionLookupTable. This is a dictionary that maps words and prefixes to integer scores. The scores are in the range -5 to 5. The dictionary was based off of data from SentiStrength[1] that we changed slightly by hand. An example of the sort of changes we made was with words such as "kill" or "revolution". For good reason, SentiStrength gave these words negative scores, but in the context of stocks, our analysis of Tweets showed that these words were often used in a positive context. Common examples of these words being used in stock Tweets are: "$AAPL is making a killing" or "The $GOOG revolution continues". Consequently, we changed the scores for these words. We decided to give the word "revolution" a positive score and the word "kill" no score (a zero score).

Furthermore, there were some very important stock related words that did not appear in the SentiStrength word list. Some very important words that we added to the dictionary were

[1] http://sentistrength.wlv.ac.uk/

“bullish”, “bearish”, “bull”, “bear”, “long” and “short”. These words appear very often in Tweets about stocks and are very helpful in analyzing the sentiment of a stock.

We also removed from the list words that have a non-neutral sentiment in regular Tweets, but are neutral in business Tweets. For example: "knife". Knife was given a negative sentiment by SentiStrength, but we removed it from the list, because in stock related Tweets, the mention of a knife is not a very good indicator that the Tweet is negative.

The word "like" was given a positive sentiment by SentiStrength and often the word like does indicate a positive sentiment, but we also found that the word "like" was also being used in the phrase "looks like" fairly often and this phrase does not indicate a positive sentiment about a stock. To deal with this case, we ignored all cases where the word "like" was preceded by "look" or "looks". When the word "like" appeared by itself, we gave the word a positive score.

Booster and negation words

There is a special family of words that we refer to as "booster words". Booster words are words that increase the sentiment of emotion words that comes after them. We also downloaded our list of booster words from SentiStrength. Booster words only had an effect on the score of words that appeared one or two words after them. An example of a booster calculation: Very very good - the double occurrence of the word 'very' adds 2 points (twice 1 point) to the sentiment of the word good (originally had a sentiment of 2). Similarly, for 'Very very bad' the combined score is -4.

Negation words are another special family of words. These words flip the sentiment of words that come after them. We consider these words also only if they come at most two words before a sentiment word. If we have two words in a row, they cancel each other. For example, "good" has a score of 2. "Not good" has a score of -2.

In our experiments we found another powerful amplifier which is the exclamation mark. We found that when at least three exclamation marks come in a row ("!!!"), they act like a booster on the sentiment of the whole Tweet.

Emoticons

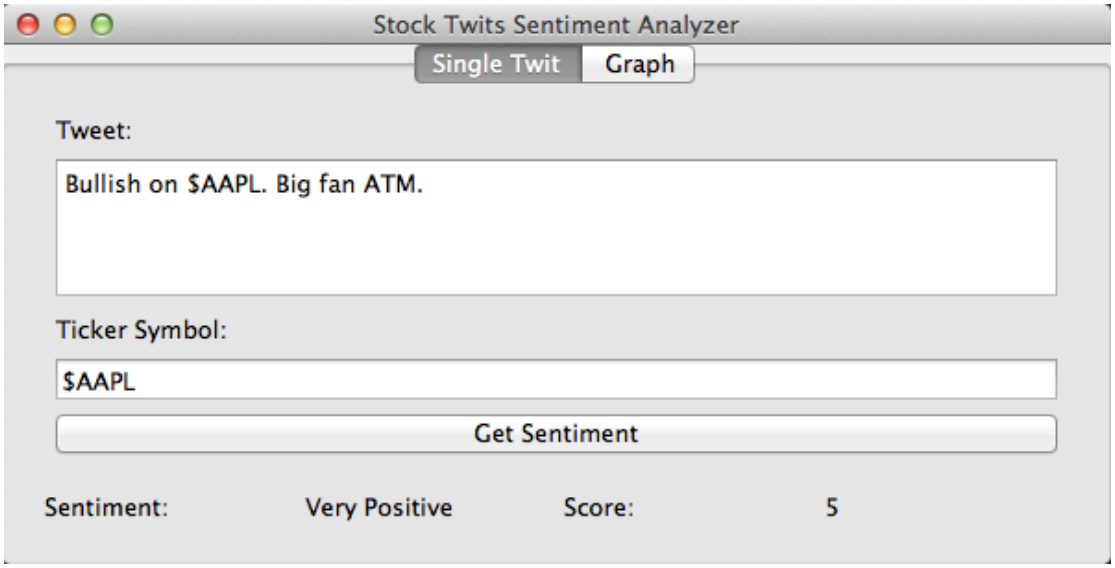
An important feature of Tweets, that doesn't exist in formal articles, is emoticons. Emotion icons are a common way for people to express their feeling and signal their sentiment. In our system, the common ':)' smiley gets a +1 sentiment.

Total score

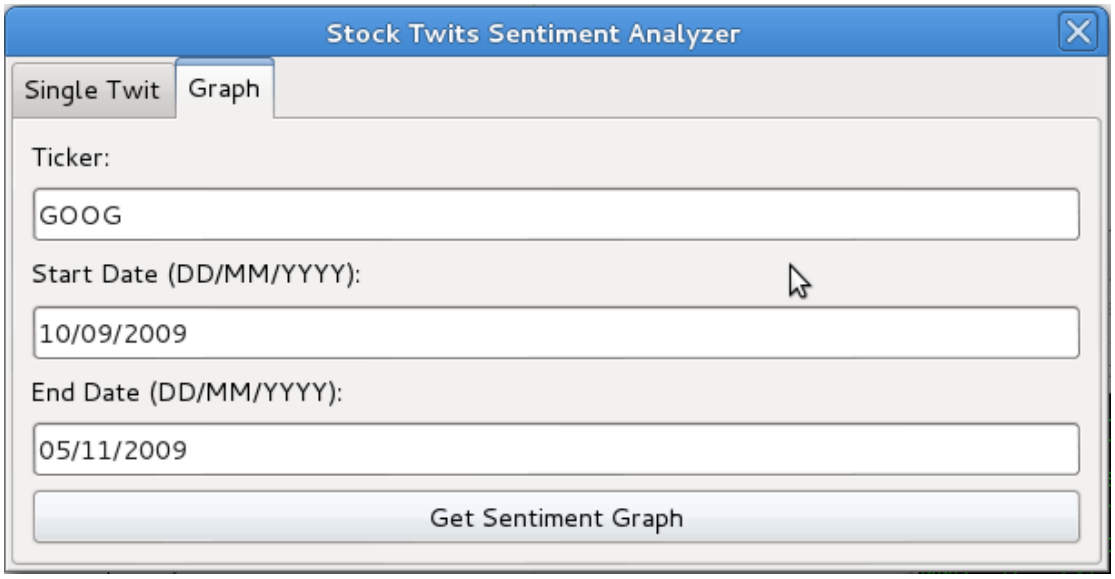
The total score of the Tweet was determined by summing all sentiment words, after all the steps we introduced.

GUI

We wrote a graphical user interface to run our program. The GUI has two tabs. The first tab allows the user to find the sentiment and score for a single Tweet. Below is a screen dump using this tab.



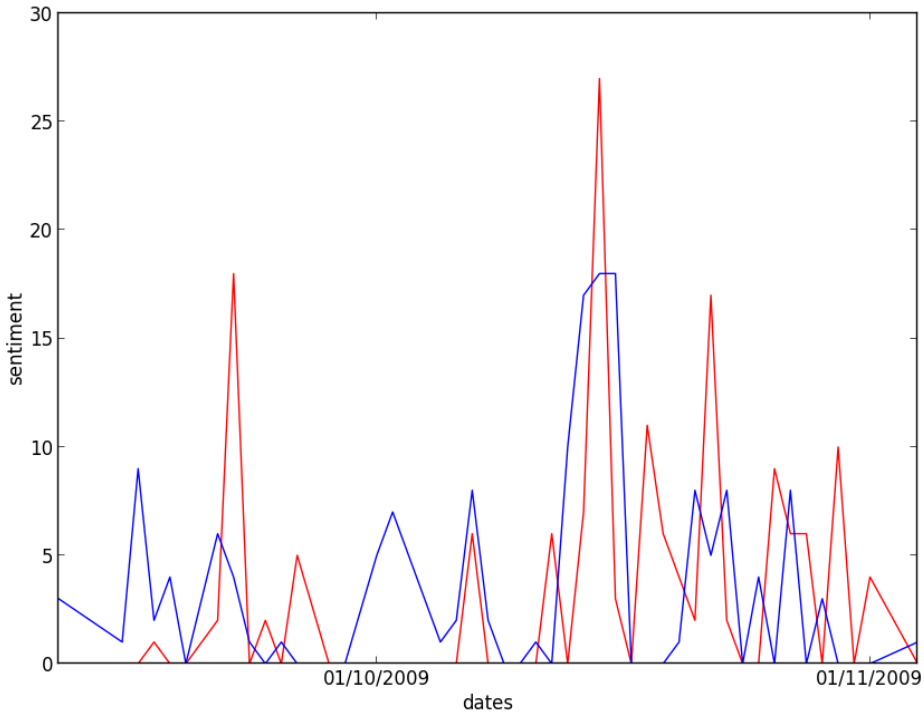
The second tab allows the user to view a graph for a ticker over a period of time. The data used to create the graphs is all the Tweets that were tweeted on StockTwits.com on publicly traded US companies from 2009 to 2011. Below is a screen dump using this tab.



When "Get Sentiment Graph" button is clicked a pop-up window appears showing a graph of the ticker's sentiment over the period of time. Below is an image showing the graph for the above input.

The blue line represents the sum of sentiment scores for positive Tweets and the red line represents the sum of sentiment scores for negative Tweets. For example, we see that around

15/10/2009 there are a lot of Tweets about $GOOG and the sum of the scores of the negative Tweets is slightly higher than that of the positive Tweets.



Final Remarks

- We attempted to analyze the stock Tweets using a database of phrases (two words or more) with their sentiment. The database originated from opfine.com which is a popular stock sentiment analysis site, based on stock related articles. We found that none of the Tweets contain any of the phrases. We explain that by the informal language used on Twitter and stock-Tweets.

- People sometimes mention specific values they believe that the stock will get to. A text mining system cannot "understand" the sentiment of such thoughts as it doesn't contain any technical details about the stock. For example our system returns: "<neutral>$GS 183.26 </neutral>". If we knew the previous value of $GS we could figure out the sentiment of such a sentence.