Text Mining Project

**Ofir Marcus 201618469**

**Eliezer Steinbok 336181490**

**Danit Lev 301643441**

**Talya Sohlberg 021987631**

Introduction

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

Sentiment analysis is one of the sub-topics of text mining. The purpose in sentiment analysis is to determine the attitude of the speaker with respect of some topic or context. Sentiment analysis is used in various fields as an analysis tool of data. As media types emerge and lots of platforms exist 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 of stocks, based on Twitter data (AKA twits). Twitter is an extremely popular platform for people to express their opinion on events or topics. It contains lots of twits written every day, each twit is under 140 characters. These twits are enough to express a sharp opinion over a topic. Assuming there is a correlation between people's opinion (and twits) about a stock and that a twit may influence other people, twitter data may be extremely valuable for trading stocks.

Our system attempts to evaluate the sentiment of a stock based on twit, and may be used to attempt to predict future fluctuations of a stock.

The data we used in the system includes 2 years of all twits written on any US publicly traded company.

The input to our system is a twit and the output is the sentiment scores for each ticker of a stock mentioned in that twit. The types are: {very positive, positive, neutral, negative, very negative}.

Analysis method

This chapter will provide technical details about the implementation of our sentiment analysis method.

The challenge in analyzing twitter data is that people write in twitter in a free language that may contain grammatical mistakes and usage of slang. Therefore, the data contains a lot of noise and cannot be easily analyzed with POS taggers or semantic parsers, as most of them assume a perfectly written text.

Another challenge twits introduce is the possibility of different sentiments to different stocks mentioned on the same twit. We solved the problem by declaring a list of 'separators', which are characters or words that mean that the context of the twit has changed, and should be splatted to sub-twits. While calculating the sentiment, we refer to the stock according to the sub-twit it's into. The separation process takes place only in case that after the separation, both sub-twits contain a ticker.

An example: '$APPL's IPhone 5 sucks. I like Nexus 4 $GOOG much more'. The separator in our case is the '.' and the analysis will be performed separately for both parts.

Next step we take is replacing all slang shortcuts with adequate normal sentences with the same meaning. For example: omg -> oh my god, r -> are, etc. The replacement is done using a dictionary called: SlangLookupTable

The sentiment analysis itself is performed using in a file called: EmotionLookupTable that contains a dictionary of words and word prefixes, each with a sentiment in a range varies from -5 to 5, all integers. The list was originally brought from SentiStrength[[1]](#footnote-1), but it was filtered by us according to our experiments, and was transferred to support business twits. In example: Revolution\* is a negative prefix according to SentiStrength in regular twits, but it has a positive meaning on business twits. We also removed from the list words that have a sentiment in regular twits, but neutral in business twits. In example: knife, kiss\*, etc. There was a special case we found in the procedure of determining word sentiments, and it concerns the word 'like'. Like has a positive sentiment when it refers to the relation of the writer to the topic, but in many cases it came after the word 'look/s' it has no meaning. We treated that case specially and ignored the sentiment of 'like' in that context.

**Booster and negation words**

There is a special family of words that we refer as 'Booster words'. Booster words are words that increase the sentiment of an emotion word that comes after them. We considered a word a as booster only if it is at most 2 words before the sentiment word. Example to the 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 2 words before a sentiment word. If we have 2 words in a row, they cancel each other.

In our experiments we found another powerful amplifier which is exclamation mark. We found that when at least 3 exclamation marks come in a row in the end of the sentence, it acts like a booster on the sentiment of the whole twit (and not on a single emotion).

**Emoticons**

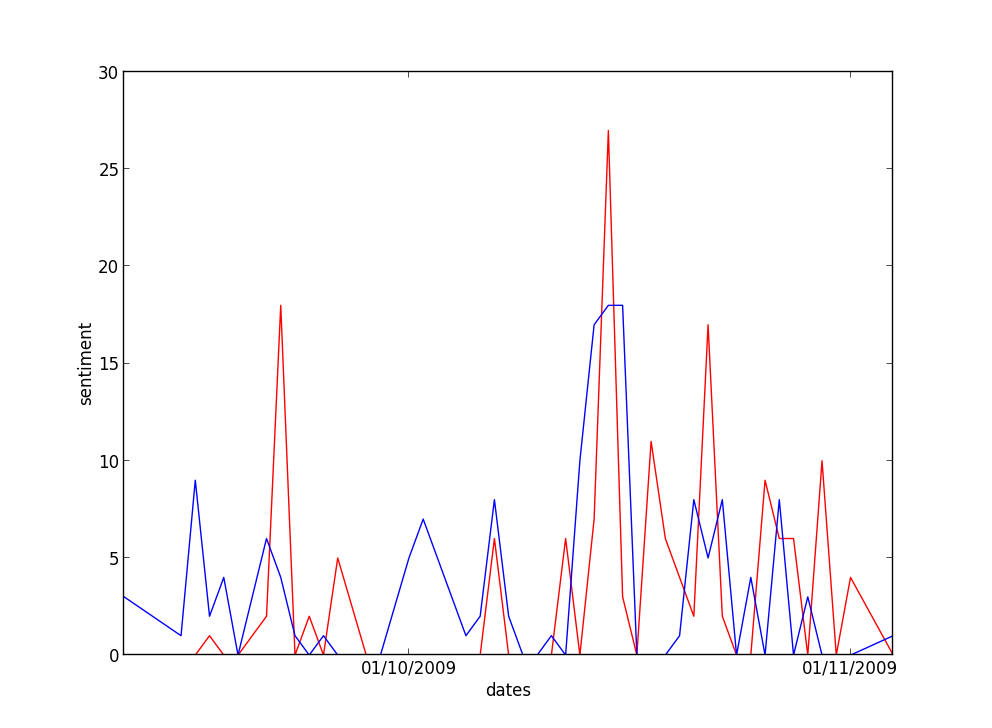
An important feature of twits, that doesn't exist in full articles, is emoticons. Emotion icons are a common way of people to express their feeling and signal their sentiment. In our system, the common ':)' smiley gets a +1 sentiment.

**Total score**

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

Demo

The red graph represents the sum of negative twits for each date, while the blue one represents the positive twits for each date.



GOOG ticker 10.09.09 – 05.11.09

Final Remarks

* We attempted to analyze the stock twits using a database of phrases (2 words or more) with their sentiment. The database is originated in opfine.com which is a popular stock sentiment analysis site, based on stock related articles. We found that none of the twits contain any of the phrases we explain that by the informal language used on twitter and stock-twits.
* 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 $GS previous value we could figure out the sentiment of the sentence.

1. [↑](#footnote-ref-1)