University College London

Twitter-based sentiment analysis influencing stock market prices

COMPGW02: Web Economics

Victor Cristian Popescu

26 April 2015

Introduction

The project aims to establish if there is a connection between stock market fluctuations and data mined from Twitter as well as whether this potential connection can be used to predict increases or decreases in stock prices. For the purposes of this project, I looked at the tweets related to Intel, IBM and Cisco captured between 13 January 2014 and 3 March 2014.

I researched the existing tools that implement sentiment analysis techniques on short sections of text and I applied a subset of them on the Twitter dataset. I also implemented a custom sentiment analysis method based on the presence of emoticons to see how that compares to the results obtained with the more established tools.

Existing tools

My research into potential methods I could use to fulfill the tasks uncovered quite a few libraries, APIs and complete Twitter sentiment analysis tools. A number of papers [1] [2] describe more complex implementations which increase the accuracy of the results, but given the time constraints, I tried to find already existing analyzers.

I found a number of complex tools built to process Twitter data, like Sentiment140 [3] and Streamcrab [4], however they only worked with real time data. I was not able to find solutions that allowed me to provide the tweets.

Next, I looked at NLP solutions, such as the Text-Processing API [5], which had the issue of free accounts being capped at 45000 requests and the TextBlob Python library. They were both based on the NLTK and they both supported built-in sentiment analysis.

Finally, trying to cover a more diverse range of methods, I also looked at the SentiWordNet [6] lexicon, which was also mentioned in the lecture notes.

Methods

I devised three ways of determining the sentiment behind a Tweet.

The first one was a rather naïve approach based on the SentiWordNet lexicon. In short, I split the tweets into individual words and looked them up in the lexicon. Each matched word's positivity and negativity was then summed up and thus a tweet-wide score was produced.

The second solution made use of the TextBlob Python library. Since it was built to analyze text, I was simply able to give it each tweet as input and the library would provide me with a tuple containing a polarity and subjectivity rating.

The third method was inspired by one of the solutions proposed by the Twitter Sentiment Analysis [1] paper and it involved looking for the presence of positive and negative emoticons in the tweet's contents. While the paper looked for only a couple of emoticons, I expanded the set to try to include the most frequently used smileys.

Implementation

The first step was to isolate the set of tweets related to each of the 3 companies I chose. I did this by using the awk tool. Each tweet that contained the name of the company (i.e. Intel or intel or INTEL) was copied in a separate file. I was able to retrieve 17703 Intel tweets, 12499 IBM tweets and 9582 Cisco tweets.

Next, I made use of a Python script I wrote. The script supported generating sentiment analysis data using either of the three methods, depending on the argument given. The "--sentiwordnet" option also required the user to provide a lexicon, so that the script could be used with updated lexicons as well (as long as they follow the same format).

The SentiWordNet method produced a tuple (positive, negative) representing the sum of those two characteristics of every word in the tweets. The TextBlob method produced a (polarity, subjectivity) tuple representing the average of the two measurements across all of the sentences in a tweet (the large majority of tweets contained a single sentence). The Smileys method produced a single number per tweet; either 1 or -1 depending on whether the tweet contained a positive smiley (":)", ":D", ":d", ";)", "=)", ":>") or a negative smiley (":(", ";(", "=(", ":<"). Tweets containing both types of smileys produced a value of 1.

The script's output was saved in JSON format as a dictionary where the dates were the keys (in yyyy-mm-dd format) and an array of object representing sentiment analysis information for each tweet (depending on the method used) as the values.

```
{
        …
        "2014-01-15": [
                        {"polarity": 0.15833333333333333, "subjectivity": 0.58333333333333334},
                        {"polarity": 0.0, "subjectivity": 0.0},
                        …
        …
}
```

Finally, the results could be viewed in line chart format using a JavaScript/HTML/CSS based visualizer I created. The charts can be seen in the Annex at the end of this document. The values plotted were the sums of the respective measurements for each day. Please note that for the SentiWordNet charts, the "subjectivity" rating was not used.

Code

The code can be found at https://github.com/victorpopescu/TwitterStockAnalyzer .

Results and conclusion

In order to be able to tell if there was a connection between the sentiment analysis charts and the actual stock market fluctuations, I used the Yahoo Finance historical market prices.

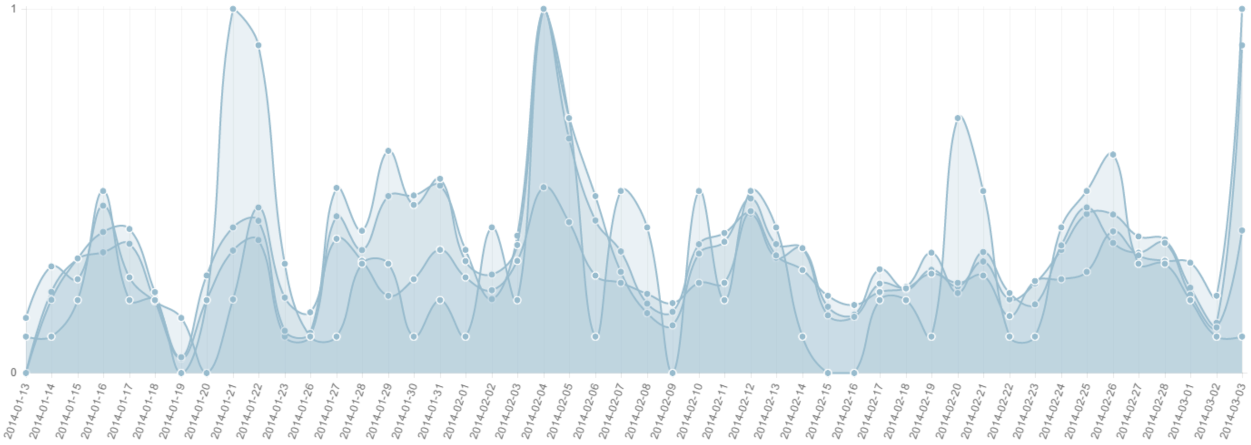


Figure 1 - Combined and normalized sentiment analysis data from all three methods I used

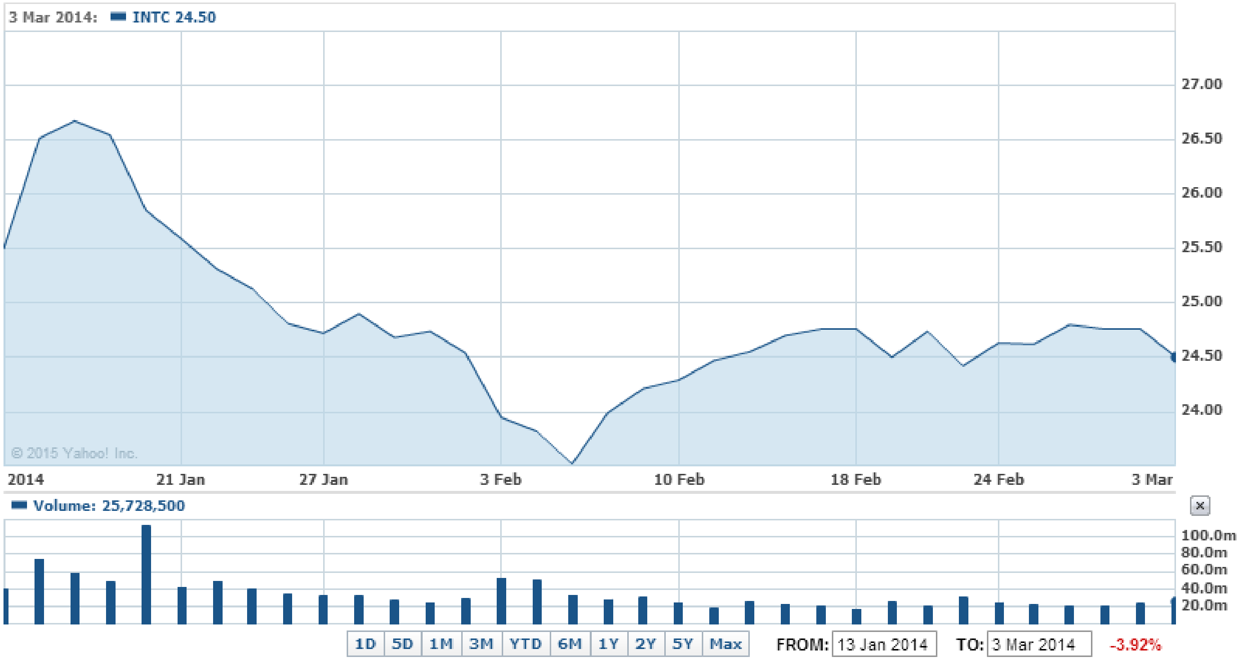


Figure 2 - Actual stock market history

It would appear that predicting stock prices based on tweets is unreliable, however one observation that I can make here is that there is certainly a connection between dramatic price changes and the people's reactions on Twitter. Intuitively, price changes are likely to cause a stir in social media, however one has to wonder if the reverse is true as well - would publicly expressed worries about a stock cause its price to drop? The chosen timeframe was too calm to draw conclusions - the chart above displays a single large drop - however, I would expect that using more complex analysis methods on a more extensive dataset would bring us closer to an answer.

References

[1] E. Kouloumpis, T. Wilson and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," Edinburgh, UK, 2011.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data," New York, USA, 2011.

[3] Sentiment140, "Sentiment140," [Online]. Available: http://www.sentiment140.com/. [Accessed 20 4 2015].

[4] cyhex, "Streamcrab," [Online]. Available: http://www.streamcrab.com/. [Accessed 20 4 2015].

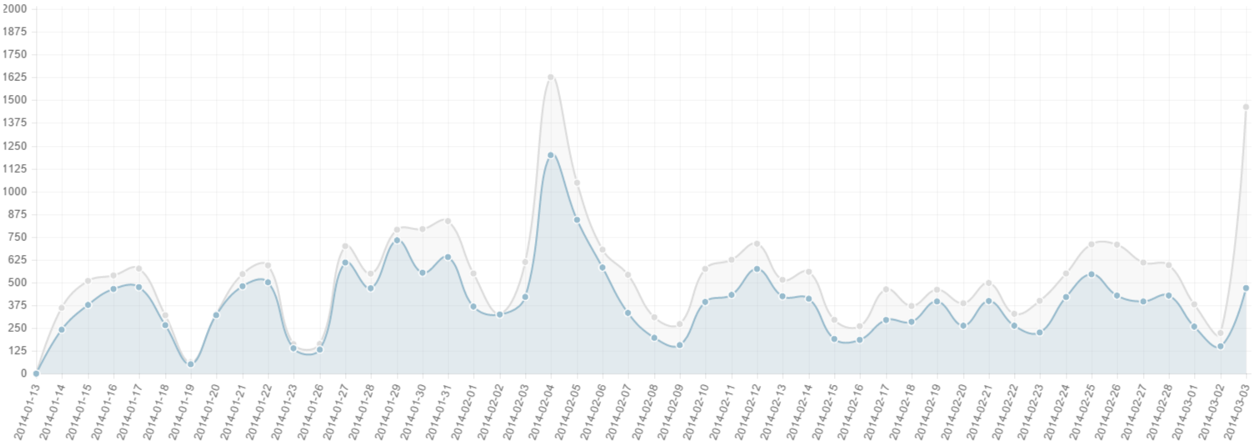
[5] japerk, "Text-Processing API," [Online]. Available: https://www.mashape.com/japerk/text-processing. [Accessed 20 4 2015].

[6] SentiWordNet, "SentiWordNet," [Online]. Available: http://sentiwordnet.isti.cnr.it/index.php. [Accessed 20 4 2015].

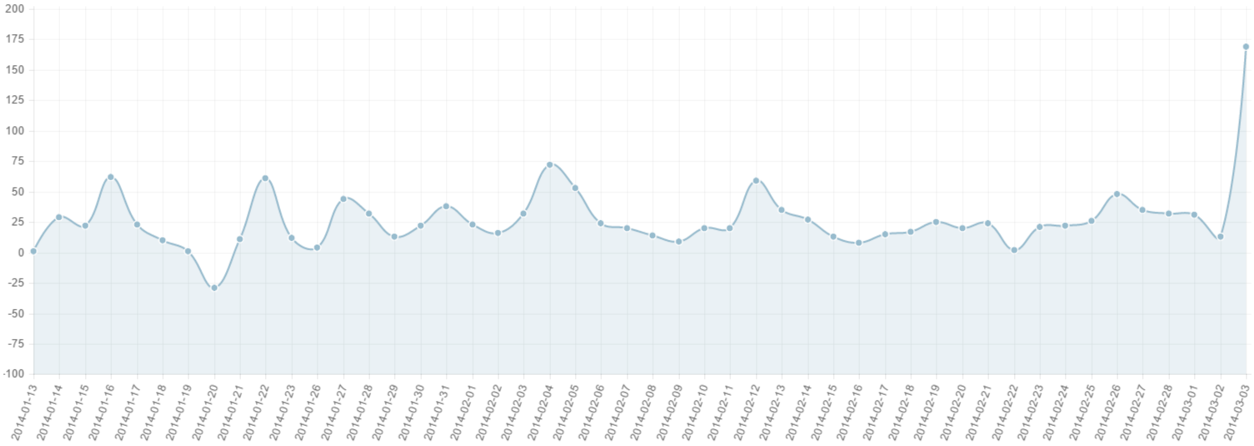
Annex

Intel

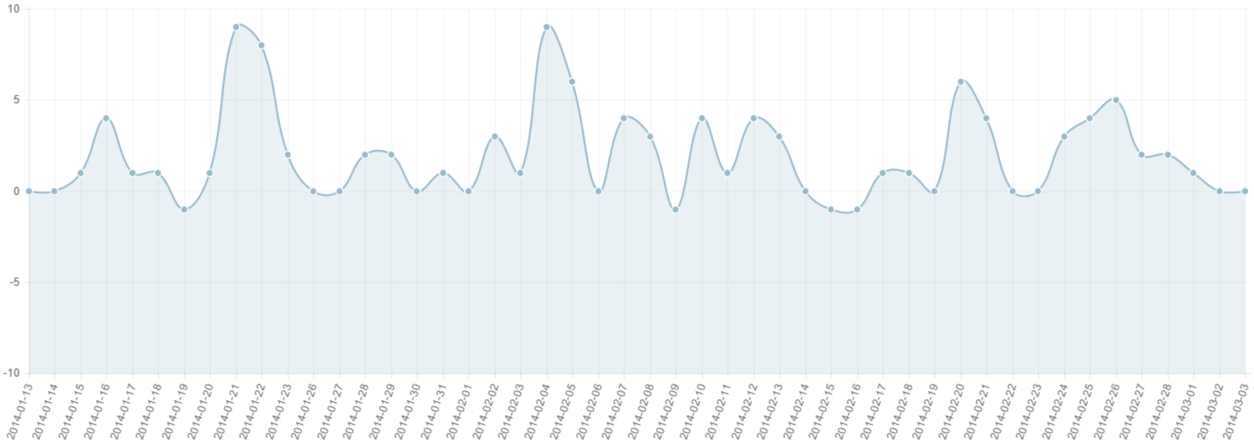
SentiWordNet



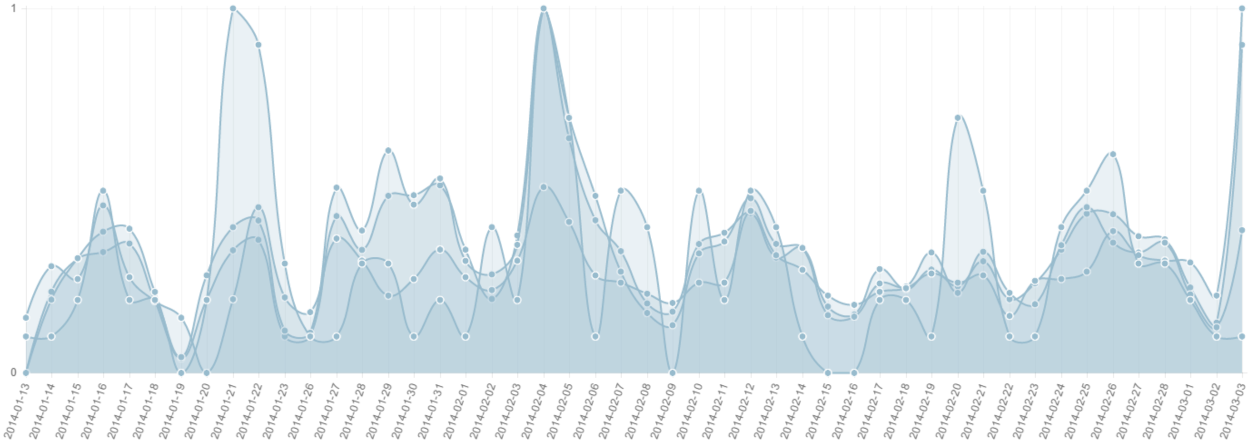
TextBlob



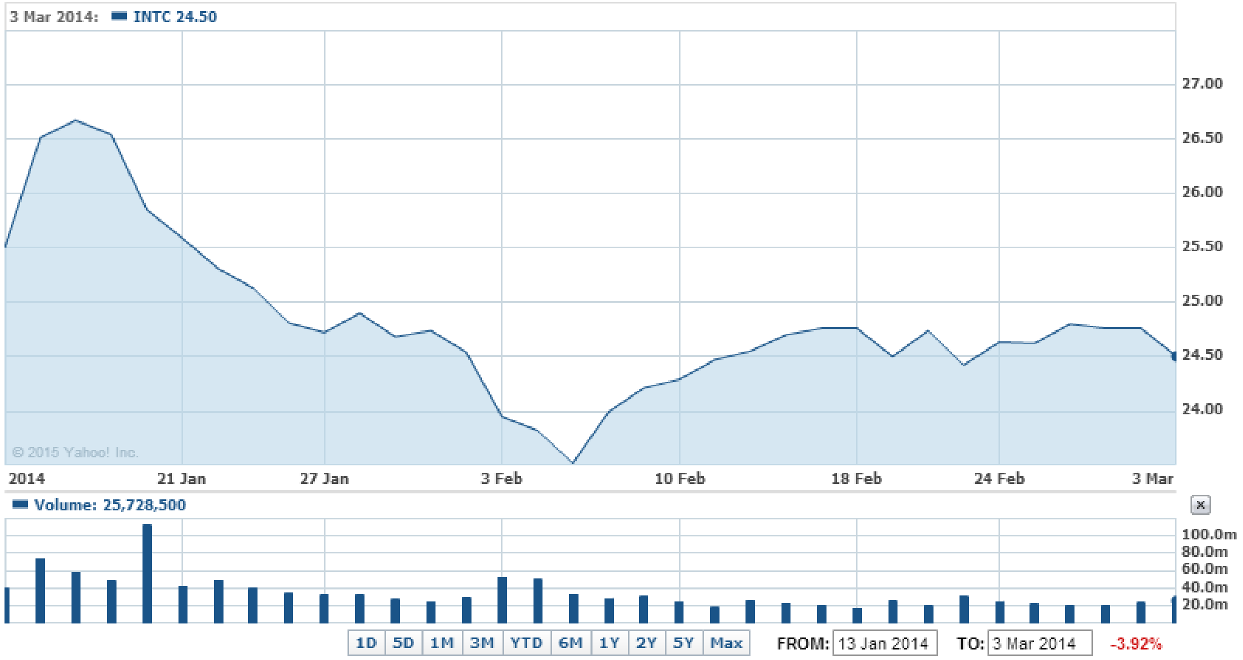
Smileys



Combined (and normalized)

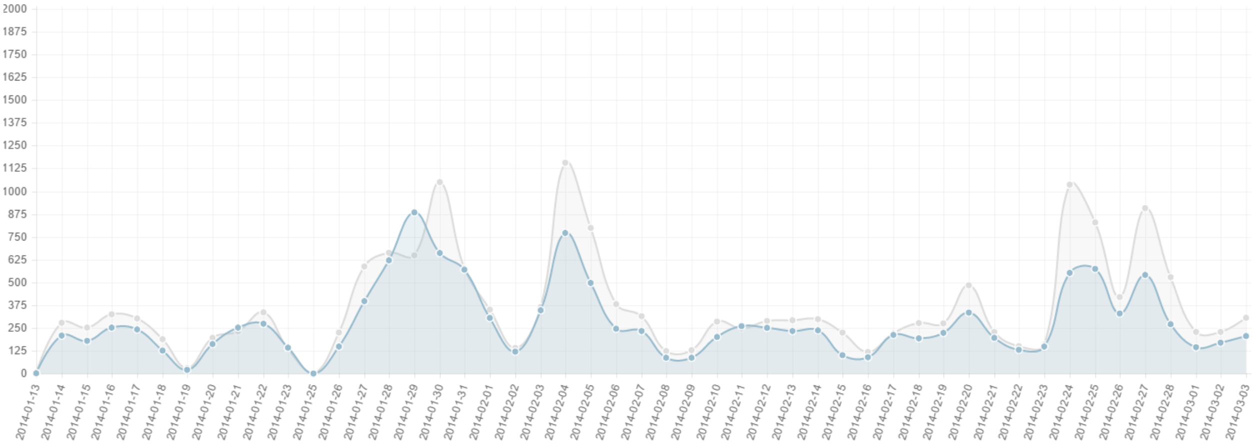


Actual price history

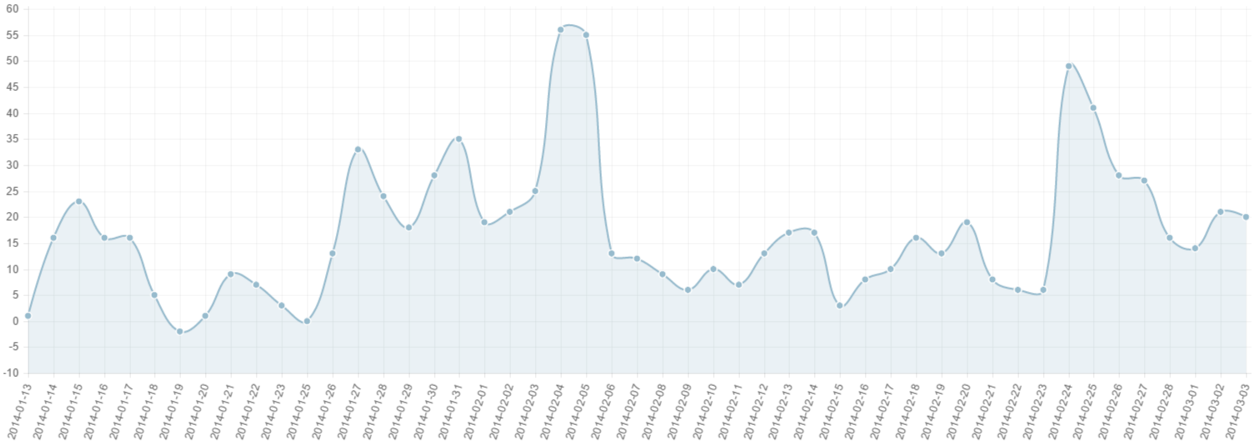


IBM

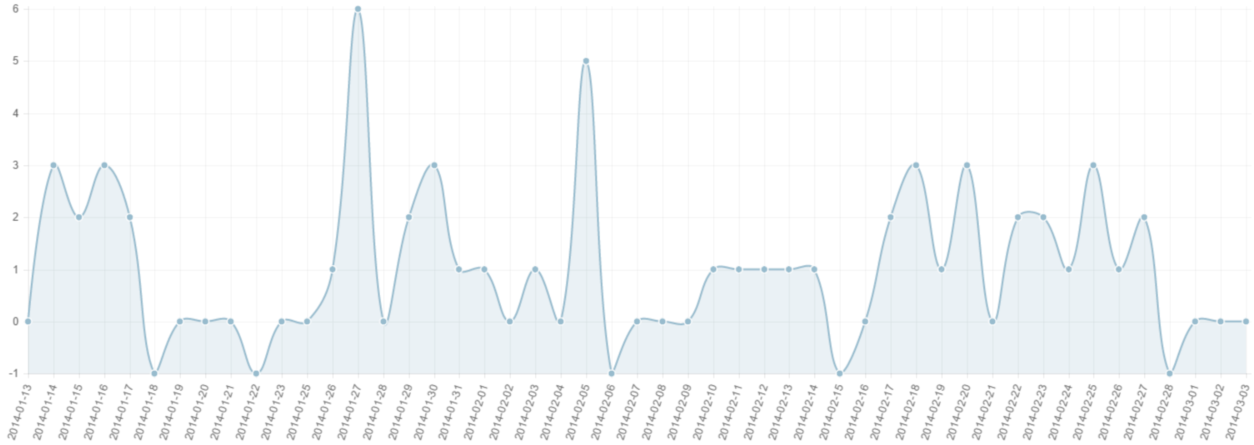
SentiWordNet



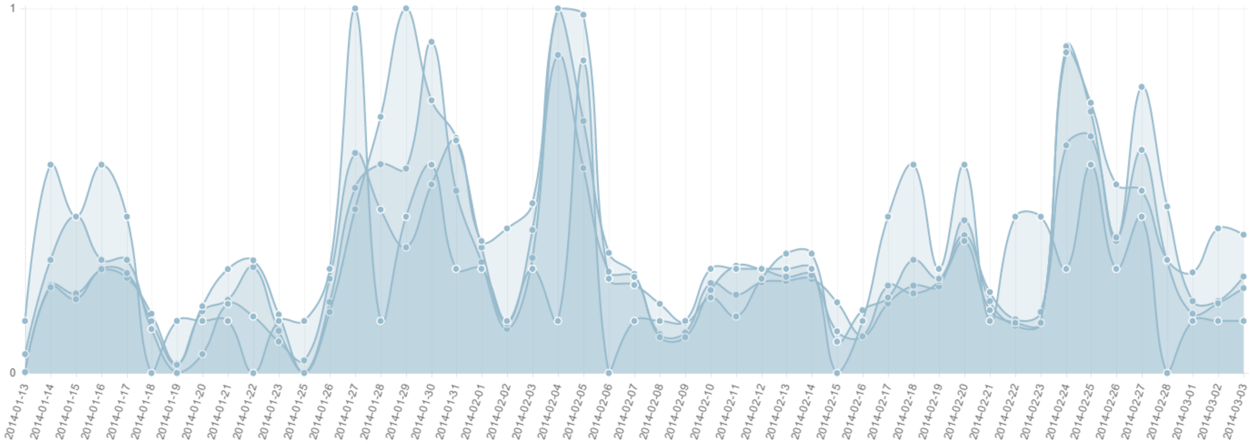
TextBlob



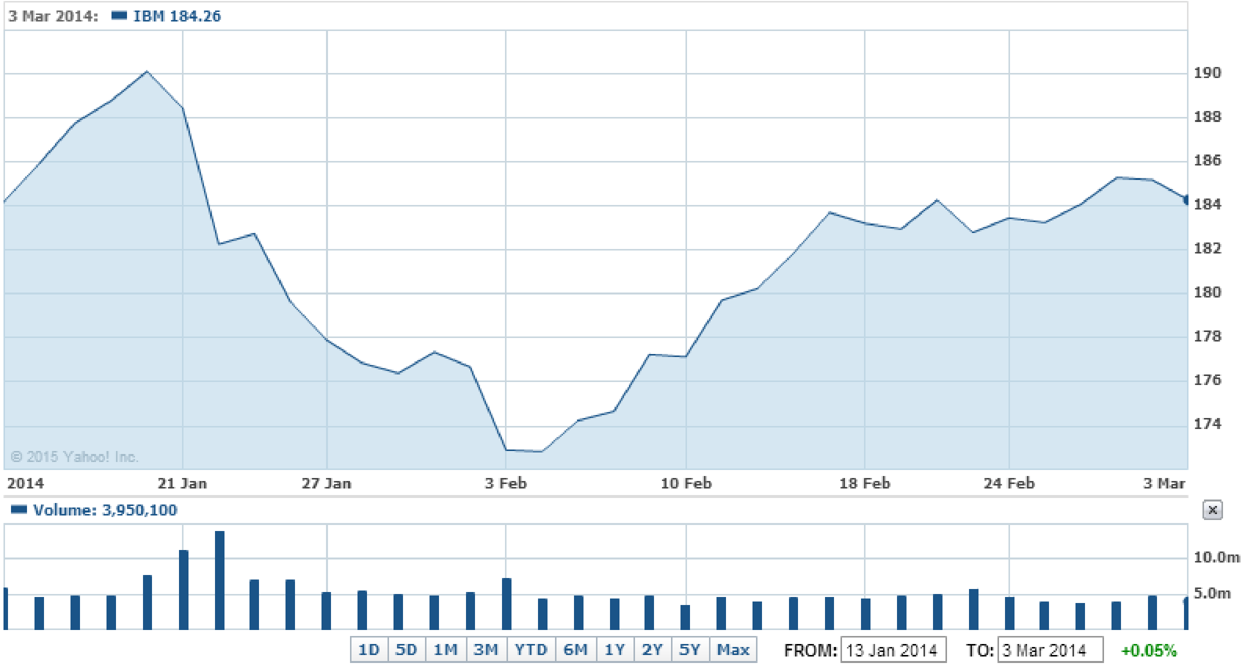
Smileys



Combined (and normalized)

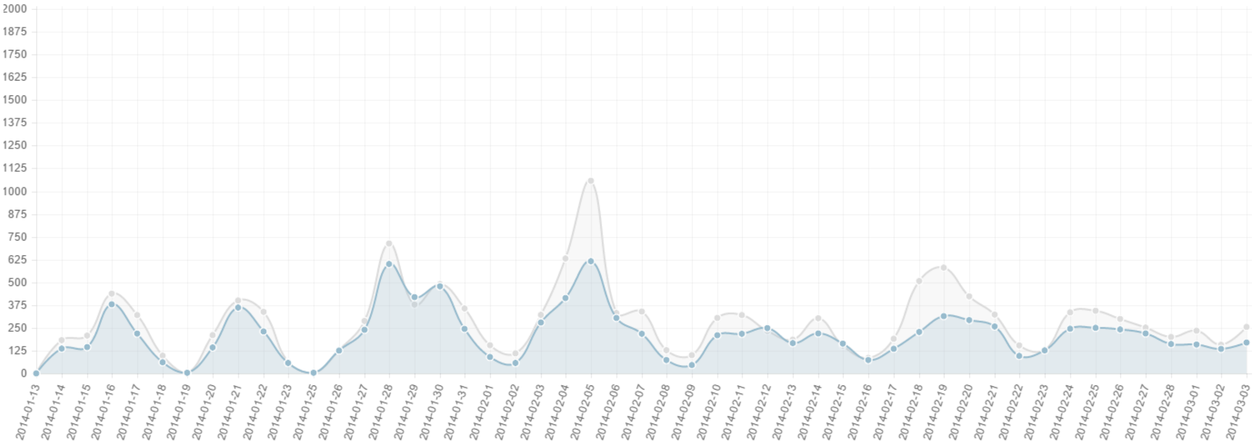


Actual price history

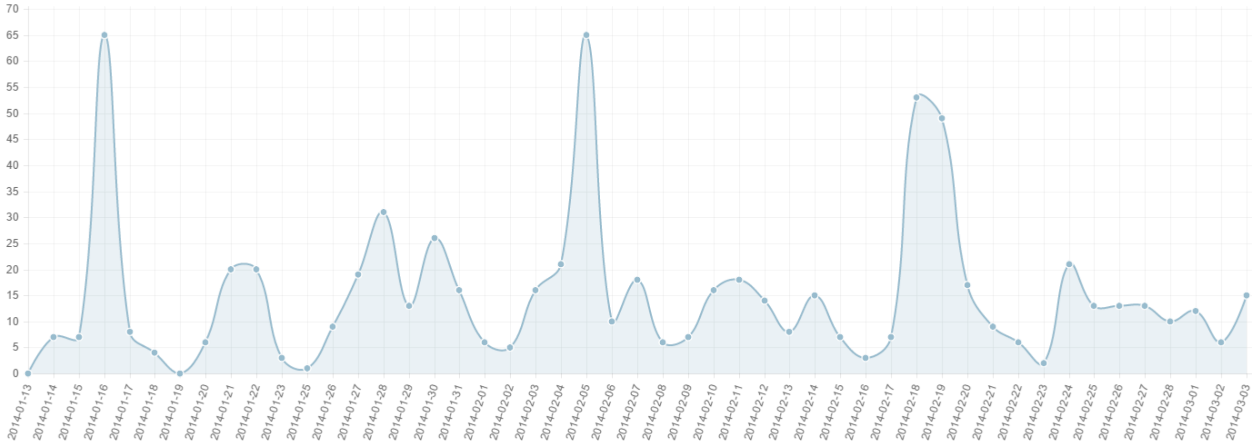


Cisco

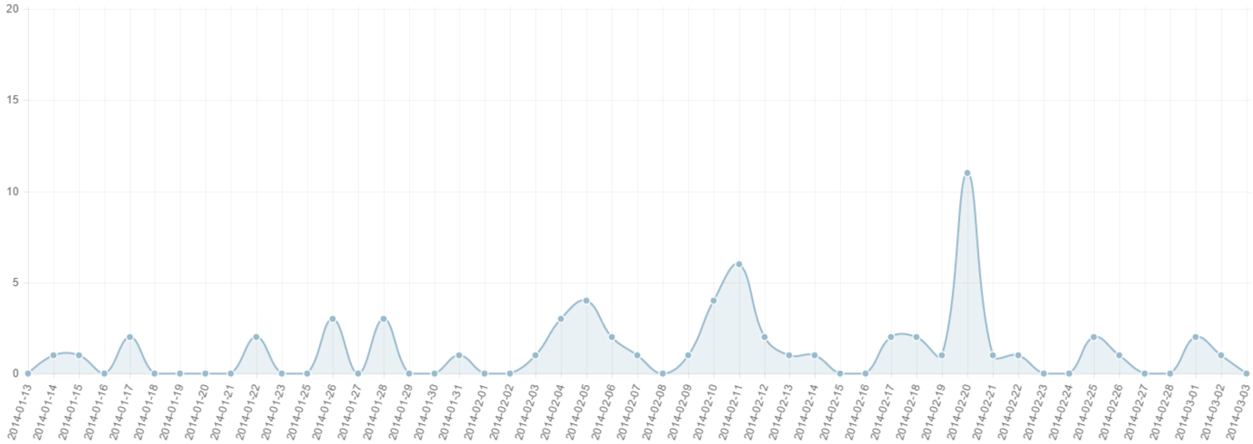
SentiWordNet



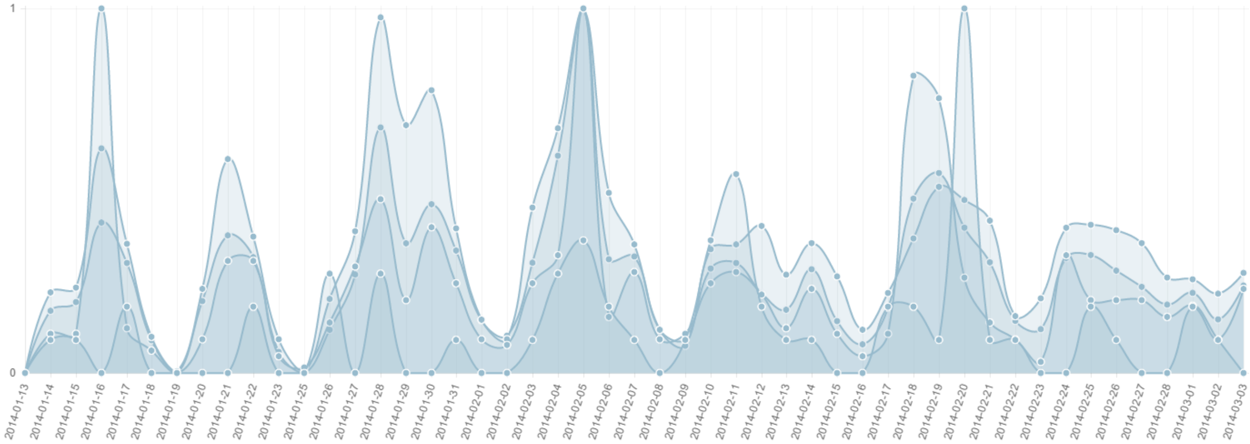
TextBlob



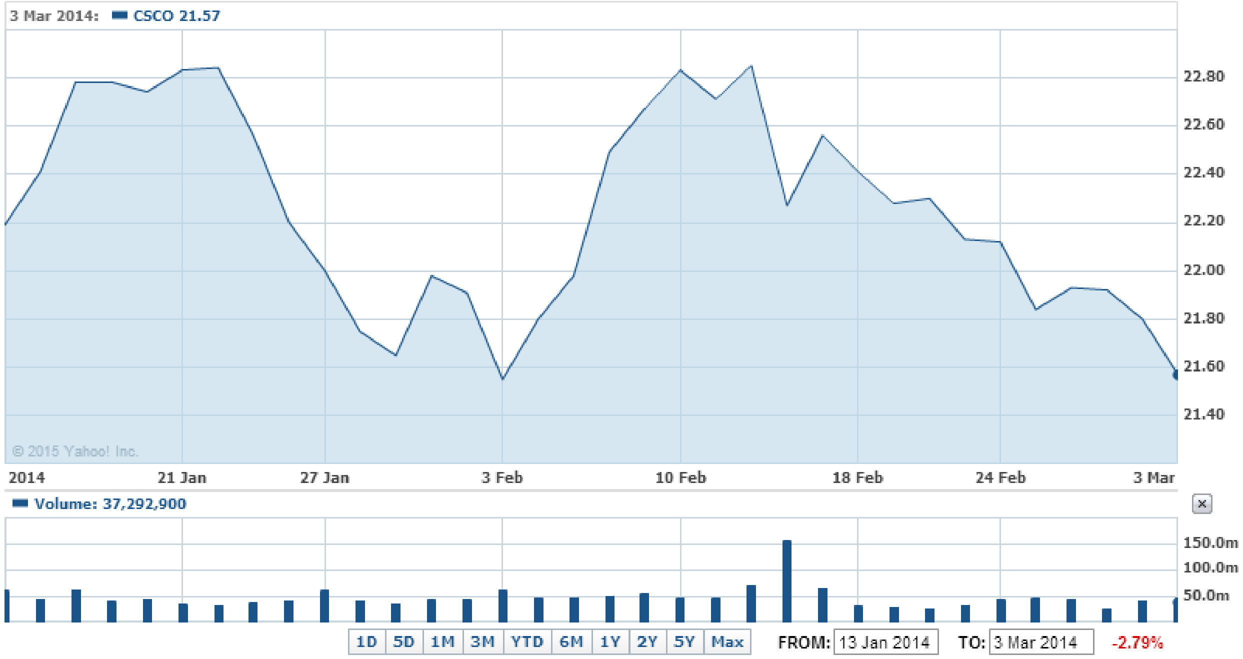
Smileys



Combined (and normalized)



Actual price history



3 Mar 2014: CSCO 21.57

Volume: 37,292,900

FROM: 13 Jan 2014 TO: 3 Mar 2014 -2.79%