Real-time Language Independent Sentiment Analysis in Social Network

Jürgen Nützel[1], Frank Zimmermann[1]

[1] 4FriendsOnly.com Internet Technologies AG,
98693 Ilmenau, Germany
{jn, fz}@4fo.de

Abstract. Micro blogging systems like Twitter aggregate 24 hours a day a huge amount of user generated content, which describes what happen around the world right now. We have developed an algorithm which extracts from these data the sentiment of many different topics. With the Twitter streaming API we access the live data feed. We filter out special regions which allow us to compare the calculated sentiment values from different regions with each other. With the live data feed we can recognize real-time sentiment changes, too. With the analyzing of the geo data of the Tweet we can determine the specific area of this sentiment changes.

Keywords: sentiment analysis, social networks, statistical text mining, smileys

1 Introduction and Motivation

Political parties, financial investors and commercial companies have all this common interest: They want to know all about the sentiments of ordinary people. Politicians want to know how their voters think about their decisions. Financial investors and commercial companies want to know how consumers like or dislike brands and their current or future products. They all have the opportunity to employ an opinion research center to ask hundreds or thousands of persons directly. But this option is expensive and time consuming. And if you want to know if sentiments have changed you have to do the polling periodically. On the other hand users of social networks create a huge amount of news, forum postings, product reviews and blogs containing numerous sentiment-based sentences. The micro-blogging service Twitter is a very special social network. 500 million[1] (active April 2012) users write in Twitter every day more than day 340 million[2] short messages (Twitter calls them tweets). These tweets are written in many different languages about almost every topic. But due to the tweet's length limitation (140 characters only) most users use smileys to express their sentiment without wasting characters. Based on these language independent smileys we calculate a statistical sentiment value for each tweet and also for each

[1] Twitter To Surpass 500 Million Registered Users On Wednesday, http://www.mediabistro.com/alltwitter/500-million-registered-users_b18842

[2] What is Twitter?, https://business.twitter.com/de/basics/what-is-twitter/

word found in the tweet. While analyzing all incoming tweets over a fixed time period of several weeks we are able to calculate for each word its average (normalized) sentiment value. The calculated sentiment value of words like “love”, “good” or “sad” is rather constant over time. The sentiment value of some other words like “weather”, “sun” or “Bahn” (the German railroad company) vary over time. Political parties, financial investors and commercial companies may be interested in the change of the sentiment value of those words or topics. Therefore it is worth to analyze their sentiment change in real-time. To do so we compared the long term sentiment values with values calculates in a shorter period (the last week or even the last day). This allows us to calculate the weekly and daily up and down of the sentiment of topics. The people’s sentiment on the weather is the most trivial result we are able to provide. Using the geo positing data of the tweets allows us to calculate regional differences in the change of people’s sentiment. All described algorithms have been implemented and tested with real data from Twitter users in the U.S. and Germany.

2 Twitter Streaming API and Geo Positioning Filtering

The algorithms we describe in this paper run on a server which is connected to Twitter via Twitter’s streaming API (https://dev.twitter.com/docs/streaming-apis). This provides us real-time access to all newly created Tweet data. As we have neither permission nor the computing power to consume all created Tweets we decided to focus on Tweets which have position data. Adding positioning data to the Tweet is optional. Therefore only 2% of all Tweets contain such data. Beside this limitation we found it worth working with this subset because it enables us to calculate regional specific sentiment values.

In our practical experiments we consumed and recorded Tweet data with positioning information from several regions. The streaming API allows filtering out Tweets by providing two coordinates which define a rectangle on the map. We defined four regions which covered Germany, New York City, parts of U.S. east coast (from Boston to Washington DC) and the San Francisco bay area.

Table 1. Analyzed regions and the amount of recorded data (until 08/26/2012)

Region	Corpus length	Effect radius	Tweets per day	Total number of Tweets until now
Germany	30 days	200km	10,800	3.1 Mio
New York City	15 days	10km	72,000	7.1 Mio
U.S. east coast[3]	7 days	140km	236,000	15.6 Mio
San Francisco bay area	30 days	20km	19,500	0.98 Mio

We record the consumed Tweets on our server using a MySQL database. This allows us also later to compare different types of algorithms. The different corpus length and effect radius are related to different number of Tweets per day and the different size of the analyzed region. In the next chapter will describe these constants.

3 Real-time Language Independent Sentiment Analysis

The goal of our approach is to detect the change of topics' sentiment values in real-time with any language specific settings. We detect smileys in Tweets to reach this goal.

Table 2. Smileys we currently use for sentiment calculation.

Sentiment	Normalized Sentiment Value	Smileys
Positive	+100	:-) ;) ;-) :P ;P ^.^ :D ;D :-D ;d -D <3 ,-)
Negative	-100	:(:-(:| :-| :~(:'-(:,(

The approach was already used by others [1, 2, 3]. The real-time demand is realized by direct connection to the Twitter API. The speed we can detect the change of sentiments depends on the amount of the data we get within a certain time period.

[3] We took the BosWash area with Washington DC in the southwest and Boston in the northeast.

3.1 Calculation of Normalized Sentiment Values

We decided to design a very simply but fast algorithm to calculate the normalized sentiment value for each word we find in the Twitter input stream.

1. We put all Tweets of the selected region (one of the four regions from table 1) in the corpus. The corpus *C* of a region contains only the Tweets *t* of a certain time span; the corpus length *l(C)*. As newer Tweets will be added to the corpus, older tweets leave the corpus. The corpus length for each region was found by experiments.
2. We calculate a sub corpus *C'* of *C*. *C'* includes only Tweets which contains a smiley from table 2.
3. For each word *w* we found in *C'* we calculate how many Tweets are in *C'* which contains *w*. We call it the Tweets with smiley of *w*: *count(w)*. If *count(w)* is smaller than a lower border (e.g. 10) the word *w* will be ignored. This means that we have too less data for a sentiment analyze of word *w*.
4. We calculate for each word *w* the value *ss(w)* (summarized sentiment) as follows. For all *w ss(w)* will be set to zero. For each word *w* we found in each Tweet *t* of *C'* we add +100[4] for a positive smiley we found in *t* and we add -100 for a negative smiley we found (see table 2).
5. The normalized sentiment value *s(w)* for the word *w* will calculated as follows:

$$s(w) = \frac{ss(w)}{count(w)} \quad (1)$$

3.2 The Variation in Time of Normalized Sentiment Values

We calculate for each word *w* from corpus *C'* every day the normalized sentiment value *s(w)*. As we got enough data from the BosWash area we could reduce the corpus length to 7 days. We have started recoding this area (and the area of San Francisco) July, 7th 2012. Until now (08/26/2012) we are able to calculate 21 sample points (which is not very much) for the variation in time of *s(w)*. If we calculate for August 7th the normalized sentiment values we use the corpus *C'* with data from August 1st to August 7th. For the 8th August we use the data from August 2nd to August 8th and so on.

We are still working on the right way to filter out noise. Currently we accept only words which are found in more than 10 Tweets which also have a smiley. To show a reasonable diagram for a word we should be able to calculate for at least 50% of the days a normalized sentiment value. For days we are not able to calculate a value we use the value of the day before.

[4] If we add or subtract 100 the end result (normalized sentiment value) will be in the range from -100 and 100.

4 Results from Recorded Data

For the area Germany we have data since August 2011. For New York we have been recording data since March 2012. This allows us to draw the variation in time with more than 130 sample points (which enough for a diagram).

For the region BoWash and the Bay Area we have started recording in July 2012. For the diagrams shown in this paper we used data from the New York area in period from April to mid of August 2012.

4.1 Different Types of Words

In table 3 we have a list of selected words. We have calculated the average value for the normalized sentiment. We also calculated the standard deviation. This shows us how much a value varies over time. Some "positive" words like "love", "amazing" and "good" have constant high sentiment values. For "negative" words like "sad" and "sick" we have not been awaiting such high deviation.

Table 3. Some words with their average sentiment value and standard deviation.

Word *w*	Area	Average of *s(w)*	Standard deviation of *s(w)*
love	New York	92.18	0.91
amazing	New York	92.49	2.68
good	New York	88.29	1.33
sad	New York	-46.61	7.69
sick	New York	-28.95	8.79
weather	New York	65.03	10.08
weather	Bay Area	81.10	2.39
weather	BosWash	58.29	5.62
wetter	Germany	76.27	5.45
vacation	New York	67.43	18.41
sun	New York	73.01	12.26
rain	New York	48.07	11.85
tv	New York	71.29	12.41
bahn	Germany	76.37	9.99
mood	New York	72.02	9.44
money	New York	65.92	10.37
apple	New York	75.43	9.02
facebook	New York	75.06	11.70
instagram	New York	77.55	14.69
ipad	New York	71.21	19.84
iphone	New York	55.32	12.11
jfk	New York	54.90	8.28
macdonalds	New York	75.87	12.74

The most interesting type of words has average sentiment values in the range of 45 and 85 with standard deviation of above 9. Like “facebook” or “instagram”.

We also saw some interesting differences in the recorded areas for the word weather (and wetter in Germany). In the Bay Area the word weather has a more positive sentiment as in the other region. Therefore it is possible to assume that this is due to the sunnier climate there.

4.2 Real-Time Change of Sentiment

The figure 1 shows the change of the words “love”, “good”, “sick” and “sad” from April to mid of August 2012. We see the high variation of the two “negative” words. We have to investigate further here. Probably some of the positive smileys from table 2 will be used ironically in negative situations.

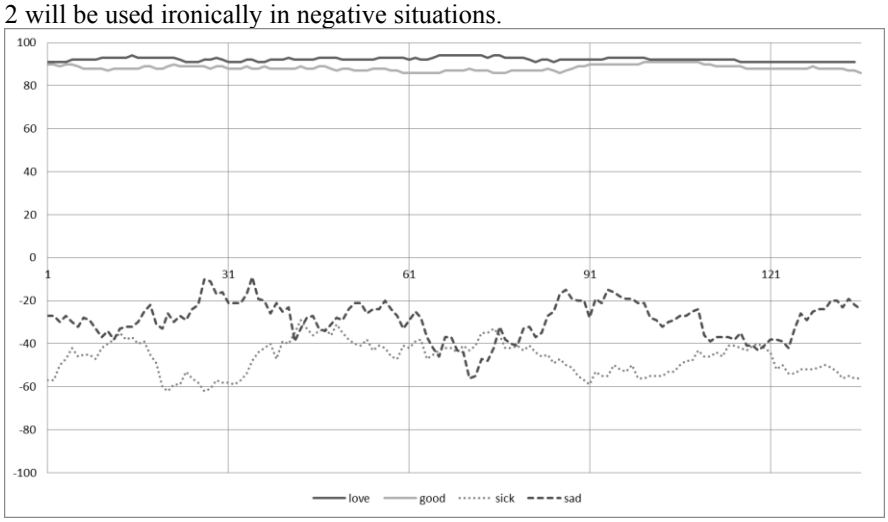

Fig. 1. The sentiment change of the words love, good, sick and sad from April to mid of August 2012 in the area New York

Figure 2 gives us a general overview of the mood change in area of New York for the period April to mid of August 2012. The sentiment value of “sun” and “rain” change most of the time in opposite directions. At the end of April and begin of May we see some correlation between “sun” and “mood”. Later the correlation is less obvious. To understand figure 2 fully we have to analyze New York’s weather data also.

Another very interesting word from table 3 is “money”. The sentiment of “money” varies a lot over time. We have to ask for the reasons also.

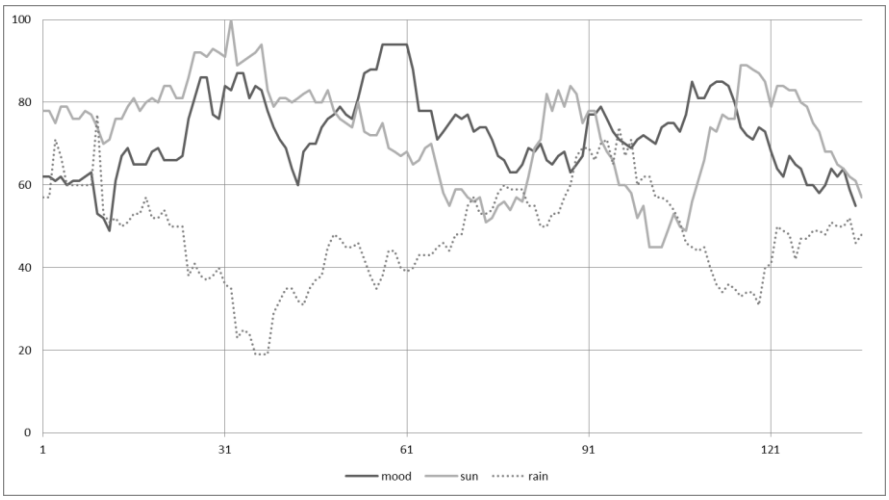

Fig. 2. The sentiment change of the words mood, sun and rain from April to mid of August 2012 in the area New York

Figure 3 compares two company names “facebook” and “instagram”. At the end of April we see a rise of sentiment for “facebook”. Is this related to upcoming going public in May? We don’t know. In the mid of July there was huge drop in sentiment for the photo service Instagram. Maybe this is related to some Twitter API limitations [4] for Instagram. We think it makes sense to investigate this further.

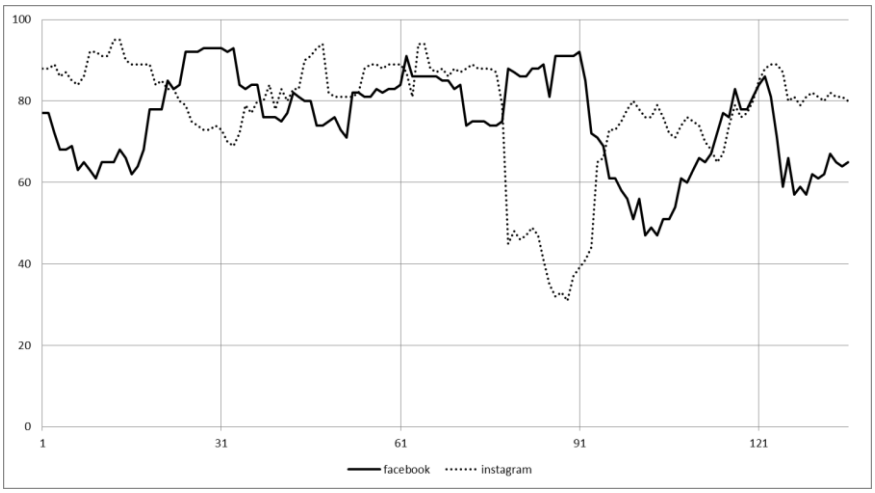

Fig. 3. The sentiment change of the words facebook and instagram from April to mid of August 2012 in the area New York

5 The Effect Radius

We decided to focus on Tweets which have GPS positioning data because it enables us to calculate local specific sentiment values. We think it is interesting to know how the sentiment differs in different areas of a country or city. In this paper we do not describe our work-in-progress approach in detail. Nevertheless we think it is worth to explain the core idea behind.

If we find in a region a Tweet with a smiley and the word *w* we do not count this word for the whole region. We limit the effect of this Tweet to circle with specific radius around the GPS position of the Tweet (the position of the writer of the Tweet). We call this radius the "effect radius" of this Tweet. In table 1 we gave different effect radius for different regions.

The effect radius defines the impact area around the word *w*. The basic idea is that the impact of a word on an observed position *x* is decreasing with the distance from writer's position from this position *x*. We have two Tweets *t1* and *t2*. Both have position data and include the word *w*. Tweet *t1* is in the near of position *x* and has a positive smiley. Tweet *t2* is far away from *x* and has a negative smiley. The Tweet *t2* has no or only a very small impact on the sentiment for word *w* at position *x*. The impact of a word on the position *x* can be described by an exponential function. The simplest exponential function is:

$$f(x) = e^{-x^2} \tag{2}$$

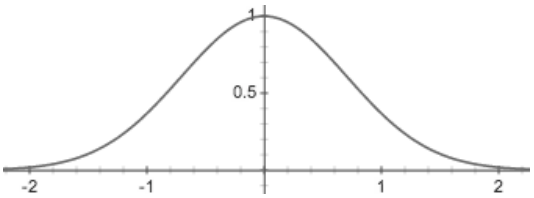

Fig. 4. Plot of the exponential function.

With an additional factor you can compress or stretch the curve and you can control the width of the curve. On the $x = 2$ position the function is nearly 0 (0.01). On this position, the impact of a word is very small and we can say it is nearly not existent. This position we call the effect radius.

6 Conclusion and Outlook

We have seen that social networks deliver us so much content that we can observe the sentiment of a large number of words. With a simple sentiment algorithm that uses the included smileys in the Tweets we have an easy and fast way to measure in real-time (from day to day) the sentiment change of words (and topics). Another advantage is

that this method is language independent. With additional position data of the Tweets we can calculate the specific sentiment value of a word on a special position. The basic idea was that the influence of a word decreases with his distance. With these new ideas we could detect local sentiment jitter in real-time. So we can see for example the current weather only by the fact, that we observe the sentiment of the word "weather".

Nevertheless many open questions and problems to solve in the future are left. On problem comes from the Tweets with positioning data. We think mobile users with the switched on positioning data option are not that type of users which normally write critical political statements. We also see improvement if we combine this approach with other methods [5, 6]. We see limitations if we work only with single words. We have to detect word pairs also. This would allow us to detect the mention of famous persons (by the detection of first name plus last name) much better.

Last but not least we want to mention that we use our platform not only to calculate the sentiment values. We are also able detect newly arising topics (words). In this approach we are also able to draw benefits from the positioning data of the Tweets. We hope to publish the work on that topic soon.

References

1. Davidov, D., Tsur, O., Rappoport, A.: Enhanced Sentiment Learning Using Twitter Hashtags and Smileys, http://aclweb.org/anthology-new/C/C10/C10-2028.pdf

2. Bifet, A., Holmes, G., Pfahringer, B., Gavaldà, R.: Detecting Sentiment Change in Twitter Streaming Data, in: JMLR: Workshop and Conference Proceedings 17 (2011) 5-11, 2nd Workshop on Applications of Pattern Analysis, http://jmlr.csail.mit.edu/proceedings/papers/v17/bifet11a/bifet11a.pdf

3. Albert Bifet, Eibe Frank: Sentiment Knowledge Discovery in Twitter Streaming Data, University of Waikato, Hamilton, New Zealand, http://www.cs.waikato.ac.nz/~eibe/pubs/Twitter-crc.pdf

4. Instagram: Twitter to Blame for Broken 'Find Friends' Feature http://mashable.com/2012/07/26/instagram-twitter-to-blame-for-broken-find-friends-feature/

5. Kubek, M., Nützel J.:, Novel Interactive Music Search Techniques, In: Proc. of the 7th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorporating the 5th International ODRL Workshop (VG'09), Nancy (2009), www.virtualgoods.org/2009

6. Kubek, M., Nützel J., Zimmermann F.: Automatic Taxonomy Extraction through Mining Social Networks, in: Proc. of the 8th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorporating the 6th International ODRL Workshop, Namur Belgium, www.virtualgoods.org/2010