Accelerate the detection of Trends by using Sentiment Analysis within the Blogosphere

Patrick Hennig*, Philipp Berger*, Claudia Lehmann†, Andrina Mascher† and Christoph Meinel‡
Hasso-Plattner-Institute
University of Potsdam, Germany
*{patrick.hennig, philipp.berger}@hpi.de
†{claudia.lehmann, andrina-mascher}@student.hpi.uni-potsdam.de
‡office-meinel@hpi.de

***Abstract*—Information about upcoming trends is considered to be a valuable source of knowledge for both, companies and individuals. A large number of market analysts working at monitoring a particular business field, with many employing manual methods to do so. Since the amount of available data on the internet is far too high for humans to monitor, which carries a major risk of substantial amount of information being missed, the necessity arose to detect emerging trends automatically.**

Weblogs are an important medium to publish information and discuss certain topics. The web platform BlogIntelligence analyzes and visualizes the content and interconnection of blogs in the blogosphere. One area of focus is the detection of trends over a period of time, which is especially helpful for product vendors.

But even more interesting, views expressed in weblog posts influence the reader's opinion. Integrating the strength and direction of expressed sentiments enhances the trend detection significantly. In this work, we introduce an approach to enrich the trend detection with sentiment analysis.

I. Introduction

An emerging trend is a topic of interest that is becoming more and more important over time. An old but often used example for an emerging trend is *Extensible Markup Language - XML* in the 1990s [1]. With the increasing amount of data that is available on the World Wide Web the need is arising to be able to detect such trends at an early stage. An very important characteristic of the World Wide Web is that opinions are getting distributed very fast.

Weblogs are important discussion platforms nowadays that serve to distribute opinions. Users can publish posts in weblogs to share information, express opinions about a certain topic, or receive feedback from other users. The blogosphere describes the collection of all weblogs and their connections. BlogIntelligence[1] is a web platform to analyze and visualize content- and network-related data of the blogosphere. The presented work in this paper is integrated into this project. The concept of BlogIntelligence consists of three different steps:

a) Extraction: In the extraction step the blogs are basically crawled. In order to achieve this a, purpose-built crawler needs to be used as traditional crawlers do not fully meet the particularities of blogs as opposed to conventional websites.

b) Analysis: The analysis step prepares the crawled data for visualization. Each blog is analyzed by multiple *Analyzers*, that process its details in certain ways. Among potentially others, there are *data analyzers* that store the meta information about the blogs into the database, *content analyzers* that store information about the content which allow content-related analyses and there are *network analyzers* that store information on the relationships and links between blogs or other communities.

c) Visualization: The last step within the BlogIntelligence framework is the visualization of the analyzed information. The Blog IntelliTrends solution is part of this last step as it provides the stored data via an interface and visualizes them in client applications.

Views expressed in weblog posts influence the reader's opinion and hence can be useful for product vendors and marketing departments. Blogs that discuss multiple products can be particularly influential because sentiments in posts and comments to a specific product can help users to decide for or against a product purchase [2]. Additionally, changes of opinion over time can be analyzed to detect patterns. If we can detect, when opinions change, further research could detect their reasons [3]. This is why we concentrate on the BlogIntelligence trend analysis explained by Hennig et. al [4] to improve it by taking care of sentiments for trends.

A term denotes an entity and can be composed of multiple words, e.g. *Barack Obama*, *Windows 8*, *Mercedes-Benz*, or *Lenovo Ultrabook*. The trend detection describes how the number of discussions about a specific term evolved in the blogosphere. The sentiment extraction describes how the opinions about a specific term evolved in the blogosphere. We focus on the term level but similarly it is possible to cluster the terms by topics to investigate trend and sentiment on topic level.

II. Related Work

A variety of approaches exists in the field of trend detection and sentiment analysis. Researchers agree that these tasks are difficult to solve and hard to evaluate.

Kontostathis et al. [5] provide an overview about semi- and fully-automatic systems to detect emerging trends in textual data. For each system they describe the concept, visualization and evaluation. The introduced projects rely on human domain expert and only a few of them use formal evaluation metrics which makes it hard to compare the systems.

[1]www.blog-intelligence.com

Goorha et al. [6] propose a system to identify emerging topics associated with terms of interest such as products, companies, or people. Therefore, they use social and mainstream media such as news articles, blog posts, review sites, and tweets. During discovery, interesting phrases near the terms of interest are extracted. Interesting phrases are used frequently, show a dramatic increase in usage, and belong to a given topic. Using this approach, Goorha et al. identify breaking concerns while filtering out one day wonders. Among others, they propose the integration of sentiment analysis to estimate the strength and positive or negative direction of sentiment.

Denecke et al. [3] analyze the evolution of topic-related opinions over time to identify interesting trends. First, they cluster blog sentences according to their topic. Then, continuous sentiment values are assigned to each topic based on the expressed opinion in each sentence. Instead of classifying sentiment keywords into groups of positive, negative, and possibly neutral sentiment keywords, SentiWordNet[2] is used here to provide continuous values representing the polarity of a single word. In addition, they propose a method to detect topics with contradicting opinions and test it on two data sets from health and politics area. Our work to enrich trend detection with sentiment analysis builds on this approach.

III. Trend Detection

The underlying idea of measuring trends is measuring the frequency of the term occurrence [3]. The trend detection, introduced by Hennig et. al [4] uses the following three criteria to calculate the trend for a term over a period of time:

- **tf-idf:** the relevance of a term in a specific post
- **#tags:** the number of times the term is used as a tag for a specific post
- **#links:** the number of incoming links for a specific post containing the term

A. Calculation

The trend calculation can be divided into three categories: incoming links, term importance and tag relevance.

1) Incoming Links: One of the most powerful structures inside the blogosphere is the inter-linkage between *blogs* or *posts*. One reason for the power of the *links* inside the *blogosphere* is certainly based on the influence that incoming and outgoing links have to search engines.

This becomes particularly important for blogs, since a lot of different types of links exist between blogs. The links placed on the blog's home page are very powerful. These links often represent other blogs containing *links* to this *blog*. Since this is sometimes shown on all sub pages this can be a powerful link mechanism.

2) Term Importance: The second aspect to look at is to analyze the content of each *post*. The different types of content are merged. Besides the real content of a *post* the database provides the *titles* and as well as the *short description* from the *feeds*. The shown indicators are based on the changed *importance index*.

[2]sentiwordnet.isti.cnr.it

In (1), the importance of a term i in a document j is calculated for a set of documents where N is the number of documents and n_i the number of documents that include the term i.

$$term_{i,j} = tf_{i,j} * log\frac{N}{n_i} \quad (1)$$

3) Tag Relevance: Since the blogosphere consists of a semi-structured format it is not sufficient to focus on the different text sources inside the blogosphere. Furthermore, the quality of the trend-detection results can be improved by including additional meta-information. The best known parts of the blogosphere structure are *tags* and *categories*. *Tags* are keywords describing the content of a *post* in a concise form. This should make it more convenient to search inside the blogosphere.

The three categories are used to determine the trend of a word. Each word is connected to a number of documents, in our case a post. Each post has a certain time stamp. By aggregating the data for the three categories for a given period of time (sliding window) and granularity, we obtain the history of the relevance as seen in Figure 1. The three categories have to be normalized to make them comparable.

Name	Degree	Intercept
Emergent Trend	positive	negative
Subsiding Trend	negative	positive
Popular Trend	positive	positive

Table 1: Interesting trend categories based on straight lines

4) Linear Regression: To detect trends an indicator has to be defined. For the detection of trends it is necessary to monitor changes over time; linear regression is perfectly suited for this work. Hennig et al. [4] made use of the categorization of Koegh et al. [7]. They are describing different time-series patterns that can be used for data mining and the meanings for them. The resulting *degree* and *intercept* determine whether a trend is ascending, descending or popular (Table 1).

5) Clustering: Users can only compare a limited number of items. Therefore, a clustering is necessary to combine semantically similar terms, i.e. belonging into the same cluster. The clusters can be created using different techniques and parameters, i.e. k-means clustering [8].

Our goal is to enrich the trend detection with sentiment information. This includes the computation of the term sentiment using sentiment keywords and the boost of the term trend. To derive the sentiment of a keyword such as *awesome*,

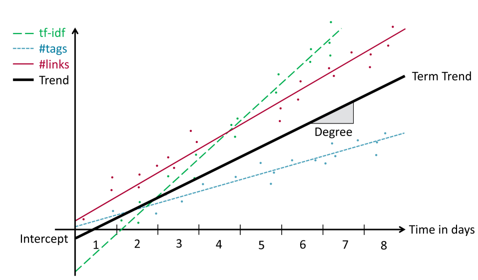

Figure 1: Detection of term trend over multiple posts using three criteria

we can use dictionaries such as SentiWordNet for the English language. Our sentiment values vary between five values with -2 depicting a strong negative opinion such as *miserable* and +2 a strong positive opinion such as *amazing*. Sentiment extraction based on keywords cannot detect irony, such as *"Amazingly, Facebook didn't do that"* and would assign positive sentiment to this example. Further, negative opinions that are expressed subtly for politeness will not include the same extreme sentiment such as *"I'm rather unsure of ..."*. The keywords should not be stemmed, otherwise *ironic* and *iron* would be mapped to the same word stem with loss of sentiment information.

IV. Concept

In the previous section, we described how term trends are built using tf-idf, #tags and #links values. In this section, we describe how this trend can be enriched by sentiment information. We derive the following three major research questions, which correspond to three steps:

- Step 1: How to assign sentiment information to terms in a post?
- Step 2: Which sentiment distributions are interesting?
- Step 3: How to boost the trend with these interesting distributions?

An important upfront decision is the impact of negative sentiments to the trend. We choose to treat negative and positive sentiments equally, to discover all terms that people spend their time thinking and writing about. Our implementation can easily be adjusted to only include positive evolution or at least terms with many positive sentiment statements.

A. Step 1: Sentiment Assignment per Post

Using sentiment keywords we can assign sentiment values to terms of interest per post. Those terms are defined through entity extraction techniques and already have a trend value. The assumption from linguistic research is that sentiment keywords occur close to those entities the author wants to describe. Following this, we define *close to* as occurring in the same sentence, such as shown in the following example:

> *"Poor Obama did the opposite but nevertheless was re-elected."*

In this sentence, *Obama* is described with the sentiment keyword *poor*. Instead of looking at individual sentences, each paragraphs can provide a broader view on one topic. The assignment per paragraph could detect multiple usages of the same entity with linguistic references such as in *"I like them"* and one ironic comment could be outvoted by many other clearly defined opinions. The downside to paragraph-based assignment are blurred sentiments when one paragraph contains multiple different opinions on various entities. Consider a paragraph that praises term A and criticizes term B, all sentiments would be assigned to both entities and could lead to a neutral sentiment. Other techniques proposed by [3] are word windows around the trend term in one sentence to assign sentiment to an entity and machine-learning algorithms. To achieve fast running time, we choose the sentence-based assignment instead of windows. We do not choose the paragraph-based model because we believe that posts have at least one sentence that includes both sentiment keywords and the entity which is sufficient.

From the sentence-based sentiment assignments we combine the sentiment of multiple sentences in one post by using their average. This leads to one sentiment of a term in post. Its value is continuous between -2 and 2.

B. Step 2: Sentiment Categories

Given the sentiment of a term for a number of posts, we can reveal interesting behavior inspired by [3]. The curve in Figure 2 shows the average sentiment value $S_{term}(d)$ of a trend term for various days d. The dots show the sentiment values of individual posts. From this distribution we can derive three interesting categories. The curve starts with a positive average sentiment because all dots are positive. After some time the overall opinion shifts to the negative side, denoted by a *change point*. If the curve mainly consists of strong opinions in either direction, we call this an *extreme sentiment*. Towards the end of the curve, the posts' sentiments largely disagree with the average sentiment due to a discussion at that time, denoted by a *contradiction point*.

a) Change point: : To detect a change point, we need to find days that are roots in the sentiment curve or where the algebraic sign changes as shown in Figure 3. For each of these days, we test the following and preceding days step by step to search for extrema in the curve with a maximum window of one week before and one week after the root. The absolute difference between these two extrema is stored for later reuse and is mapped to the range between 0 and the upper bound 2. We only consider those cases where the extrema lie outside an ϵ-area, in our case outside of ± 0.5.

b) Contradiction point: : For each day in the curve, we calculate a contradiction value and it's maximum will serve as the contradiction point. The contradiction value C of a day d is given as follows:

$$C_{term}(d) = \frac{Stddev_{term}(d)}{0.2 + |S_{term}(d)|} \times W_{term}(d)$$

$$W_{term}(d) = \frac{\tanh(\#\{posts\ at\ day\ d\} - 3)}{2} + 1$$

The factor $W_{term}(d)$ is a weight function and the factor $Stddev_{term}(d)$ is the standard deviation of all posts at that day

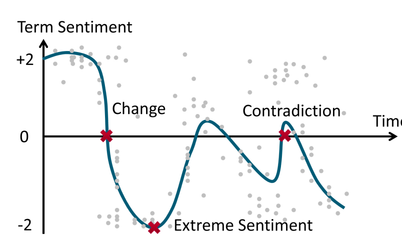

Figure 2: Three sentiment categories based on the average sentiment of posts

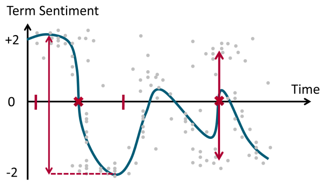

Figure 3: Change and contradiction point for opposing opinions either consecutively or simultaneously

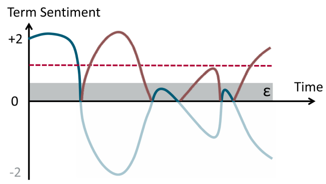

Figure 4: Average of extreme opinions as trend impact if above the average sentiment of all curves

d. The intuition of the contradiction value is shown in Figure 3 where multiple opposing opinions will lead to an average sentiment close to zero and a high variance of the individual posts at the same time. By combining $Stddev_{term}(d)$ and $S_{term}(d)$ in this way, the contradiction value rises as the average sentiment becomes closer to zero and the variance increases, too. We adjusted the formula proposed by [3] to our data and our need to effectively limit the upper bound: We choose the standard deviation instead of the variance and in the denominator we use $|S_{term}(d)|$ instead of a square and add 0.2 to avoid zero in the denominator. The weight function $W_{term}(d)$ increases the contradiction as the number of different posts for a day d increases with a range between 50% to 150%. Consider a day with five posts or more, then $\tanh(+2) \approx +1$ so that $W \approx 1.5$ which increases the contradiction value considerably. Another day with only one post will have $\tanh(-2) \approx -1$ so that $W \approx 0.5$ which decreases the contradiction value. This formula might exceed the range of 0 to 2 in extreme cases, therefore we cut of all larger values later and assign the value 2.

c) Extreme sentiment: : We add the third category extreme sentiment to increase the number of terms with interesting sentiment points. We calculate the average of all absolute sentiment values to describe how extreme the opinions are expressed over the entire time period as shown in Figure 4. If the result for this curve is above the average of all other curves, it will be considered in the next step. The average of all terms, the *mean average sentiment*, should be calculated only once and be reused as threshold for all terms. This is called a *semi-static algorithm* as is requires one run over the data to preprocess this threshold and will consider this threshold static afterwards [9].

C. Step 3: Trend Boost

For one given term, we can combine these three categories to one trend boost number. We choose the maximum of these three categories as it is sufficient if one category criterion is fulfilled. To find one representative change point and one contradiction point, we use the maximum from these categories respectively. The third category already produces only one number, which will be downgraded to 75% because it can be found for many terms.

The final trend boost shall be interpreted as percentage, hence all values need to be mapped to the range between 0 and 1. As described above, all three categories are in the range between 0 and 2, so we can easily map them to a percentage.

Given Trend

RANK	WORD	DEGREE
53	HBO	26.2
58	Jews	24.0
110	America	11.5
160	Muslim	8.0
170	Iran	7.2
177	Gaza	6.3
270	Muslims	3.1
303	Hamas	2.9
304	Islam	2.8
~~341~~	~~Muhammad~~	~~2.1~~
503	Amazon	1.1
511	Deutschland	1.0
595	president	0.6
~~754~~	~~August~~	~~0.4~~
760	President	0.4

Trend Boost

Adjusted Trend

RANK	WORD	DEGREE
28	HBO	53.0
30	Jews	50.1
34	Muslim	43.8
36	Iran	43.0
37	Gaza	42.0
41	Muslims	38.9
42	Hamas	38.7
43	Islam	38.6
45	America	38.3
50	Amazon	36.8
51	Deutschland	36.8
52	president	36.4
53	President	36.1
54	Friday	35.9
55	Barack Obama	35.7

Figure 5: Top 15 boosted terms through sentiment analysis ranked by their trend degree

Contradiction values that exceed the limit 2 are cut of the curve at this point. The trend boost can be calculated as follows:

$$
\begin{aligned}
Boost(term) &= 0.5 \times max\{max\{change\ points\ of\ term\},\\
&\quad min\{2, max\{contradiction\ points\ of\ term\}\},\\
&\quad 0.75 \times extreme\ sentiment\ value\ of\ term\}
\end{aligned}
$$

$$
\begin{aligned}
degree_{new}\ (term) &= degree_{old}(term)\\
&\quad + Boost(term) \times Stddev(\{degrees\})
\end{aligned}
$$

With this trend boost, the given trend can be updated. We boost the trend by changing the degree and not the intercept. Boosting the intercept could alter an emerging trend with negative intercept to become a popular trend with positive intercept instead. Emerging trends are far more interesting and the sentiment should not have any impact on popular versus emerging trends, as their difference is in the number of posts. The degree cannot impact the categories of emerging or popular trends, but it provides a ranking among these trends. To boost the degree we use the old degree and add the previously defined $Boost$ percentage of the standard deviation in all degrees to it.

V. RESULTS

To showcase our results from this implementation we will present the 15 top trend terms that we boosted, show the impact of each category and present the sentiment curves of one example term. Finally, we discuss the difficulty of creating a gold standard.

A. Data Size and Top 15 Terms

We process 119k posts from BlogIntelligence that were published in autumn 2012. These posts contain 2700k sentiment keywords with 93k distinct sentiment keywords, which leads to an average of 23 sentiment keywords per post. Further, entity extraction techniques retrieve an average of 27 entities per post that might be assigned sentiment. From the same set of posts, 150k distinct terms are assigned a trend value. As we need terms that have a trend value and a sentiment assignment, the joined set contains 38k terms of relevance, from which 20k terms occur in three or more posts.

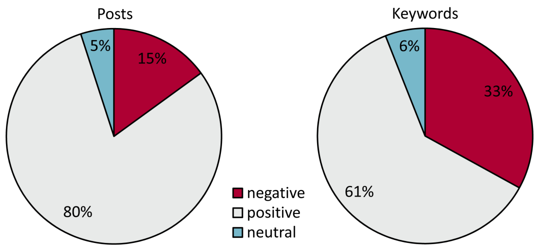

Figure 6: Distribution of positive, negative and neutral posts and keywords

The trend of the 20k terms is boosted by our sentiment analysis and Figure 5 illustrated the trend boost of the top 15 terms ranked by their degrees and shows their trend rank before the boost. We were not able to derive a sentiment assignment for all 150k trend terms, which is why the top boosted term is ranked at position 53 in all trend terms and boosted to rank 28. Further, this table shows that some word groups such as *Muslim, Iran, Gaza, Hamas, Islam* and *president, President, Barack Obama* are assigned similar ranks. We assume that they co-occur in many sentences and hence are assigned similar trend and sentiment values. The latter group overtakes the term *America* which does not get boosted as much. Terms such as *August* that people usually do not have extreme opinions about do not appear in the top 15 boosted terms. We can explain the sentiment for *Friday* with the opinions for *Black Friday*.

With our investigation, we can answer the question, whether people write more positive over negative posts. Figure 6 illustrates this percentage and indeed 80% of all posts have a positive sentiment average. Similarly, more positive keywords were discovered in these posts, a total of 61%. There is still improvement in the used sentiment extraction technique. It is biased towards positive keywords, [3] also state that their automatic approach assigns values that are slightly more positive than their gold standard. This might be due to more subtle criticism between the lines instead of extreme attacks. Further, our sentiment keyword data set maps some phrases to two opposing sentiment values, e.g. *insanely popular* is extracted with two sentiment values -2 and +2.

B. Categories

In Section IV, we introduce three categories to find interesting trend terms and how to calculate them from the average sentiment per day. An example sentiment curve is shown in Figure 7 for the term *Obama* taken from posts in late 2012. The average sentiment oscillates between -2 and 2 and the number of the posts per day increases towards the end. Days with multiple posts can have posts with varying opinions and therefore the contradiction increases at these days. The contradiction curve has its peeks at those points, where the average sentiment approaches zero. Change points might appear, where the average sentiment crosses the time-axis in a window of 14 days. In this example, the maximum from the first category *change* is 1.7, the maximum from the second category *contradiction* is 2 and the third category *extreme* is 0. The third category is not represented in this graph as it is merely the average of the sentiment curve. In this sample data set, the average is 1.05 which is not outside the epsilon area of 1.2.

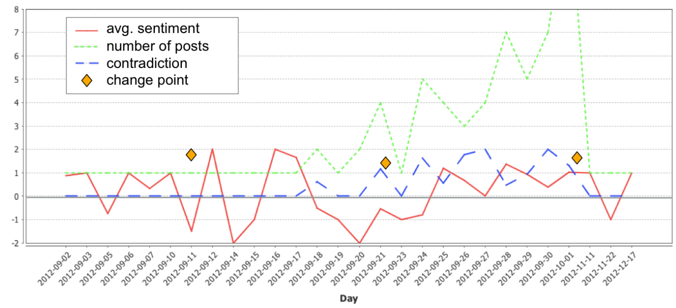

Figure 7: Curve of the trend term *Obama* with contradiction and change points

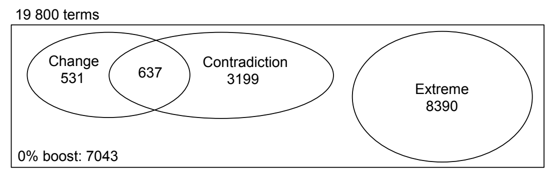

Figure 8: Frequency of impact on trend boost for each category

The third category capturing extreme opinions needs the *mean average sentiment* as threshold. This notion requires two averaging steps on top of each other, first averaging the sentiment of every term and then averaging this outcome. It is computationally faster to average all values in one step which gives more weight to terms with more posts. The resulting threshold does not change much tough: instead of 1.249 the faster average is 1.237 and therefore we use the faster version.

In Section **??** we describe how the category for change points is implemented in two ways: imperative and declarative. During the inspection of the terms that were found by the SQL view variant but are not found by the originally intended imperative way, we see terms that have a change point within a two-weeks windows but the gap between these days is larger than one week. By inspection of the terms that the SQL views are missing we see only terms that have a change of sentiment within one week, caused by one post of the opposite opinion. When this post is the only post on this day, then the imperative algorithm will find it. We decide that the SQL variant is better in this case, as it uses the maximum and minimum values of the entire week.

The final trend boost per term is calculated by choosing the maximum from the three categories per term. Figure 8 shows the number of times that each category is the strongest. We consider only terms with three or more posts. For 7043 out of 19800 terms no category applies, i.e. no interesting sentiment points are found. For 8390 terms, the extreme category is the maximum, although it is downgraded to 75% of its value, because of this huge number of terms. The intersection of the change and contradiction category contain mostly elements with 100% boost in both categories.

C. Gold Standard

Creating a gold standard in trend detection is an extremely difficult task. There is no common notion of one term being a higher trend than another term, as this is very subjective and largely depends on the person's background and interests. The question whether a term is a trend or not, is easier to answer, therefore we should focus on the trend categories.

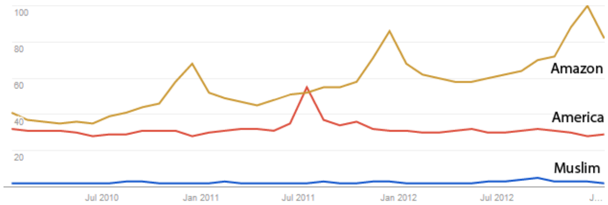

Figure 9: Occurrences of keywords in search queries over time visualized by Google Trends

To evaluate the quality of our results we inspected some samples manually as it is extremely hard to create a gold standard. The first idea to generate a gold standard is crowdsourcing the task to Amazon Mechanical Turk[3] and let users define the trend of terms on a scale from -2 to +2 to allow a ranking and additionally let them decide on emerging versus popular trend. We can derive a gold standard for sentiments by using Facebook[4] likes, but we do not have dislikes. A similar idea is Swipp[5] which allows users to rate everything on a scale of five stars. In both cases, we can compare how much users like something but there is no temporal information and both ideas really capture the sentiment of users instead of the trend. A gold standard for sentiment extraction is already created by [3]. They hired four annotators to assign sentiment values at sentence level and average their results. These results vary a lot which shows the difficulty of sentiment assignment. They do not have a gold standard for their detection of interesting sentiment distributions with contradiction and change.

Most of all, we want to evaluate the trend of terms instead of the sentiment, i.e. not only comparing absolute numbers such as Facebook likes, but rather inspecting the evolution over time. Google trend[6] shows how often a term is used in search queries over time. Google presents the top 10 terms of popular and emerging trends in adjustable time periods, but there is no overlap with our top k terms as we have very different data sources. We are left with the test, whether two terms in our small result set have the same ordering in the larger data set from Google trends. In Figure 9 we inspect three terms taken from our top 15 terms with Google trends between 2010 and early 2013. We see that the term *Amazon* occurs more frequently in search queries than *America* and *Muslim* and that the usage of *Amazon* increases compared to *America*, describing a strong trend.

Similarly, in our top 15 boosted terms, *Amazon* is boosted more than *America* to increase it's trend degree. Both terms can be seen as popular trends, as both started with high search volumina in 2013 compared to *Muslim*. The downside of Google trend is the dynamic y-axis with the maxima always denoting 100. In Figure 9 we cannot decide on the strength of trend described by the *Muslim* curve, we need to zoom in with Figure 10 to see that it is a popular trend increasing from 40% to 50% during three years. Both curves of *Amazon* and *Muslim* increase, and both terms are boosted largely in our top 15 terms.

[3] www.mturk.com
[4] www.facebook.com
[5] www.swipp.com
[6] www.google.com/trends/

Figure 10: Zooming in on a less frequent term with Google Trends

VI. Future Work

Future investigations could evaluate improvements of our described sentiment assignment to trend terms. The authority of a weblog, such as their PageRank, could be used as weights. The first and last sentence in each post could contribute with a higher weight to the sentiment of a term in a post. Instead of evaluating trends as time changes, our method can easily be adapted to evaluate different trends over space, because different blogs are published in different countries with different domains [3].

VII. Conclusion

In this paper we showed how the trend assignment to terms can be enriched by sentiment analysis from weblogs. The definition of trend terms is a difficult task and depends on a combination of a high number of occurrences and the increase of occurrences. We proposed a mean to add the opinion of weblog authors to this calculation with the intuition that positive and negative opinions should be treated equally. Therefore, we detect interesting sentiment distributions with three categories: change of sentiment over time, contradiction at one point in time and extreme sentiment as average. The implementation with an in-memory database on the BlogIntelligence data set shows promising results in running time and quality. These results reveal that the expressed and extracted overall sentiment in posts is slightly positive.

References

[1] A. Kontostathis, L. Galitsky, W. M. Pottenger, S. Roy, and D. J. Phelps, "A survey of emerging trend detection in textual data mining," 2003.

[2] C. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1, p. 235.

[3] K. Denecke, M. Tsytsarau, T. Palpanas, and M. Brosowski, "Topic-related sentiment analysis for discovering contradicting opinions in weblogs," *Technical Report*, 2009.

[4] P. Hennig, P. Berger, and C. Meinel, "Identify emergent trends based on the blogosphere," in *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, vol. 3, Nov 2013, pp. 41–45.

[5] A. Kontostathis, L. M. Galitsky, W. M. Pottenger, S. Roy, and D. J. Phelps, *A Survey of Emerging Trend Detection in Textual Data Mining*. Springer, 2004.

[6] S. Goorha and L. Ungar, "Discovery of significant emerging trends," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010, pp. 57–64.

[7] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," *Data mining in time series databases*, vol. 57, pp. 1–22, 2004.

[8] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[9] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York., 1999, vol. 463, p. 177.