Domain-Specific Identification of Topics and Trends in the Blogosphere

Rafael Schirru, Darko Obradović, Stephan Baumann, and Peter Wortmann

German Research Center for Artificial Intelligence (DFKI)
Knowledge Management Department
Kaiserslautern & Berlin, Germany
{schirru,obradovic,baumann}@dfki.de
p_wortma@cs.uni-kl.de

Abstract. Staying tuned to the trends and opinions in a certain domain is an important task in many areas. E. g., market researchers want to know about the acceptance of products. Traditionally this is done by screening broadcast media, but in recent years social media like the blogosphere have gained more and more importance. As manual screening of the blogosphere is a tedious task, automated knowledge discovery techniques for trend analysis and topic detection are needed.
Our system "Social Media Miner" supports professionals in these tasks. The system aggregates relevant blog articles in a specified domain from blog search services, analyzes their link structure and their importance, provides an overview of the most active topics and identifies general trends in the area. For every topic it gives the analyst access to the most relevant articles. Experiments show that our system achieves a high degree of sound automated processing.

1 Introduction

Besides the traditional media such as newspapers, radio, and television the World Wide Web (WWW) plays an increasing role as an information source for the shaping of public opinions. With the expansion of the Web 2.0 a shift in user behavior can be observed. More and more users are no longer only consumers but they become also producers of content. They contribute and comment pictures, videos, and bookmarks in resource sharing platforms, write reviews in online shops, and collaboratively collect information in wikis. In our article we focus on blogs as a medium that allows every user of the WWW to easily express her opinion about anything.

In this context, we collaborate with professional market researchers. For them it is essential to stay tuned about reviews and the acceptance of products, or about trends in the area of their interest. Traditionally, this is done by screening broadcast media, but in recent years, social media like the blogosphere have gained more and more importance for the evaluation of products and trends. A good indicator for this is Microsoft's PR action from December 2006, when they sent free Vista laptops to influential bloggers.[1] However, the large amount of available information sources, the problem to obtain a good overview of them, and the difficulty to rate their importance makes the monitoring of social media a tedious task if performed manually.

We developed a system named "Social Media Miner" to support professional market researchers in these tasks. There are three key actions that are automated by our system, which are illustrated in Figure 1. In the first action, the system aggregates relevant blog articles of a domain from different blog search services for maximal reach. In this data set, it analyzes the structure between the blogs and articles for ranking purposes. In the second action, we monitor the number of articles for each day. For periods of four days we extract the topics of the articles by applying textual data mining techniques. That way we provide an overview over the discussions in the domain. A user who is interested in a topic can select the most relevant terms from its label and in a third action, the system returns a ranked list of relevant articles as reading recommendations for exploring the selected topic.

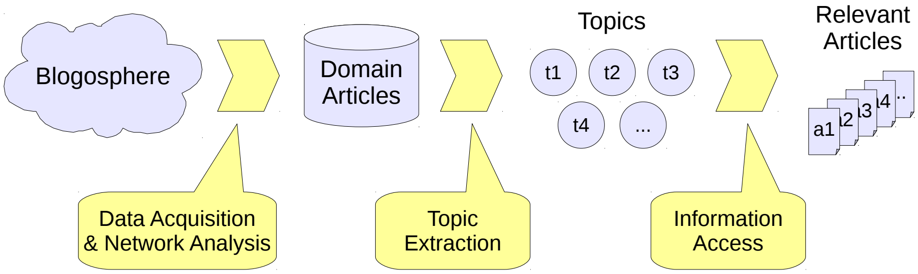


Fig. 1. System workflow.

The remainder of this article is structured as follows. In Section 2 we present related work in the fields of trend analysis and topic extraction. In Section 3 we describe how blog articles are aggregated from different search engines. Next, in Section 4 we briefly explain how networks are generated out of the data that allow us to derive social authority metrics for the articles. We depict the topic extraction process in Section 5 and the process of information access to articles of a topic in Section 6. Then we go on to present a first evaluation of our system in Section 7 before we conclude our findings and present our ideas for future work in Section 8.

2 Related Work

Research in the blogosphere can be roughly divided into two categories. The analysis of its structure and the analysis of its content. The structural analy-

[1] `http://apcmag.com/microsoft_sends_ferraris_to_bloggers.htm`

sis focuses on the ranking of blogs, the identification of communities and the dynamics of the blogosphere, mostly with Social Network Analysis (SNA) methods. The contentual analysis instead focuses on the blog articles and investigates what bloggers are writing about, usually with respect to a time dimension.

As explained in the introduction, we are following the second direction and investigate the current topics in the blogosphere. The analysis of its structure is however an essential tool for understanding the field and for ranking blog articles. Kumar et al. [1] describe how information evolves in the blogosphere, namely in "bursts" of increased activity. This increased activity does not necessarily imply that there is more real information around, but merely reflects its popularity.

In the area of contentual monitoring of the blogosphere, there exists the common idea of trend detection based on keyword frequencies, i. e., how often a certain keyword appears throughout all articles. The search service BlogPulse[2] implemented an automatic trend discovery for weblogs [2]. It is capable of identifying key persons or key phrases in the whole data set on a daily basis. This form of global monitoring is a very good indicator for the blogosphere as a whole, but it provides only insights for social researchers and curious individuals, hardly for market researchers interested in a specific domain, or any other focused observers. For these tasks, BlogPulse offers "trend search". Based on a query, it plots the number of matching articles per day over a certain time period. This gives a good impression of the general popularity of a domain or product, and periods of increased activity or buzz. You can also compare the trend lines of two different queries for a comparison of activity. This methods can provide deeper insights for specific keywords, but it fails to explain the curves any further.

The same trend lines are offered by other blog search services as well, be it commercial ones like Icerocket[3] or more research-oriented ones like BlogScope.[4] This method is currently well-established and state-of-the-art and practice.

To detect the topics in the corpus of blog postings our system uses algorithms from the domain of topic detection and tracking (TDT). TDT is concerned with finding and following new events in a stream of documents. In [3] the following TDT tasks have been identified: First is the segmentation task, i. e., segmenting a continuous stream of text into its several stories. Second, there is the detection task which comprises the retrospective analysis of a corpus to identify the discussed events and the identification of new events based on online streams of stories. Third is the tracking task where incoming stories are associated with events known in the system. In this work we focus on the detection of topics in a corpus of blog postings.

In [4] Schult and Spiliopoulou consider the problem of finding emerging and persistent themes in accumulating document collections which are organized in rigid categorization schemes such as taxonomies. They propose ThemeFinder, an algorithm for monitoring evolving themes from accumulating document collections. The algorithm works as follows: In the first period, it clusters all documents

[2] `http://www.blogpulse.com/`

[3] `http://www.icerocket.com/`

[4] `http://www.blogscope.net/`

in the collection. In the following periods, it clusters the new documents with the old feature space and compares the new clusters to the ones found in the previous period. If the clusters of two adjacent periods are similar with regard to their themes and if the quality of the clustering is not declining significantly, then the original feature space is kept. Otherwise a new feature space is build for the documents of the latest period and the next comparison. Thematic clusters are represented by a label, consisting of a set of terms that have a minimal support in the associated cluster. Thematic clusters that survived over several periods, despite re-clustering and changes of the feature space, will become part of the classification scheme. The authors put special emphasis on the evolution of topics over time. Our system deals with data from the Web 2.0, where many topics emerge in a short term and decay just as quickly. For that purpose we currently only focus on the topic detection task. Our approach combines statistical analysis (publication trend) with topic extraction techniques.

3 Data Acquisition

In order to find blog articles relevant to our domain, we define the appropriate keywords for a search query and regularly aggregate the search results from multiple blog search services. That way, we do not have to set up a complete search engine infrastructure by ourselves, and we can reach more articles than a single search service can offer, as our experiment will show.

3.1 Blog Search Service Analysis

In a preliminary step, we evaluate the quality and reach of five popular blog search services. These are Technorati,[5] Google Blogsearch,[6] Bloglines,[7] Icerocket and BlogPulse.

As the domain for this test, we have chosen the keyword “Henrietta Hughes”, which unequivocally refers to an event on February 10th 2009, where this homeless person talked to US president Barrack Obama. The event had a major impact in broadcast media, as well as social media, especially the blogosphere.

Using this search query with the aforementioned services two weeks after this event, we aggregated and manually verified a total of 871 unique blog articles writing about this event. The most important finding concerns the percentage of articles each search service contributed to the aggregated article set. The best service, Icerocket, reached 51% here, while the other search services range between 21% and 35%. Thus, by aggregating results from multiple services, we can acquire a significantly larger data set as the basis for our analyses.

Concerning the validity of the search results, we discovered a number of unreachable sites, non-blog articles as well as presumably related pages that do not even mention the lady’s name. Apart from Google Blogsearch’s results, where

[5] `http://www.technorati.com/`

[6] `http://blogsearch.google.com/`

[7] `http://www.bloglines.com/`

only 51% were valid, i. e., blog articles on topic, the validity of the remaining services is between 84% and 93%.

Consequently, we left Google's service out of the final aggregation component, and implemented a number of heuristics, based on the URL, meta-data and the site content, in order to filter out as much of the invalid results as possible.

3.2 The Aggregation Component

For our analyses, we need the URL of each blog article along with the date of publication, the title and the textual content. All search services allow to return the query results sorted by date, enabling us to exactly fetch the results in our individual time period of interest in the first step of the aggregation process. In a second step, each result is validated by the RSS entry on the blog site, fetching the accurate date, the full title and the available textual content.

Another important building block of our data is the link structure among these articles. We want to track all links, where the textual content of an article is referring to another current blog article in the domain. In most cases, the link targets will be articles already existing in the data set, but eventually, this will discover new relevant articles to be added to our data set. These links are used later as a social assessment of relevance and authority of articles, as widely known from PageRank [5] and similar methods. We impose some requirements on these article links, in order to include only expressive ones. First of all, links between articles on the same blog are ignored, since their expressiveness of authority is doubtful at best.

In a next step, we extract the underlying blog URLs out of the article URLs and gain a second type of data, the blogs. We then collect the blogroll links between these blogs, according to our method presented in [6]. These will serve as additional authority indicators in the subsequent network analysis.

4 Network Analysis

Our acquired data enables us to use SNA methods [7] to derive social authority values from the link structure.

Our data set can be represented in two networks that are linked with each other. We have a first directed network of articles, in which the nodes represent the blog articles and the edges represent the links between these articles. We also have a second directed network of blogs, in which the nodes represent the blogs in our data set, and the edges represent the blogroll links between them. These two networks are connected via a relation between the blog articles and their originating blog. These relations can be used to map metrics from one network to the other one.

We are interested in an authority value of blog articles for the reading recommendations on detected topics. First of all, we apply the PageRank algorithm [5] on the article network to obtain initial authority values for the articles. Alternatively, Kleinberg's HITS algorithm [8] can also be chosen for this task by the user. This authority is grounded on the fact, that often-cited articles are more likely to contain original and interesting content than less- or non-cited ones.

Second, the blog in which the article has been published is also a very strong indicator for the authority of an article, as the public usually trusts into a blog or blog author, not into a specific set of her articles. Therefore, we also calculate an authority value for each blog from the link structure of the blog network.

The final authority value of an article is the mean value of its initial authority from the article network and the authority value of its blog. That way, by the application of established network ranking algorithms, we can provide an importance indicator to the reader, when she has to choose articles from a given list. In our application this metric is used to sort the article list of a topic, so that the authoritative articles are listed first.

5 Topic Extraction

We identify the topics in our corpus of blog articles by applying textual data mining techniques. First we aggregate articles that were published within a time interval of four days. The time window is shifted by two days in each iteration of the algorithm, i. e., first time window from day 0 to day 3, second time window from day 2 to day 5, and so on. We will refer to these iterations of the algorithm as *runs* subsequently. The setting has been chosen as a typical use case, where a market researcher requests every two days an analysis of the domain for the last four days. However the size of the time window can be adapted to the needs and preferences of the user as well as to the publication volume in the respective domain.

The process steps of our topic extraction algorithm are depicted in Figure 2 and will be described in detail subsequently:

Data Access We use the titles of the blog articles as the input for the topic extraction algorithm, as they are considered to reflect the content of the associated articles appropriately in the majority of the cases. On account of the usually large number of articles per topic, a topic can still be detected reliably even if some titles do not perfectly describe the content of the respective articles. We also conducted experiments including the full text of the articles, however, the best clustering results were achieved when only the titles of the blog articles were used.

Preprocessing We convert the terms contained in the titles to lower case characters, remove punctuation characters and stop words. Further stemming is applied to bring the terms to a normalized form. We use the Snowball stemmer[8] for this purpose. The normalized profiles of the blog articles are represented according to the "bag-of-words" model, i. e., they are represented as vectors where the features correspond to the terms in the corpus and the feature values are the counts of the words in the respective articles.

[8] `http://snowball.tartarus.org/`

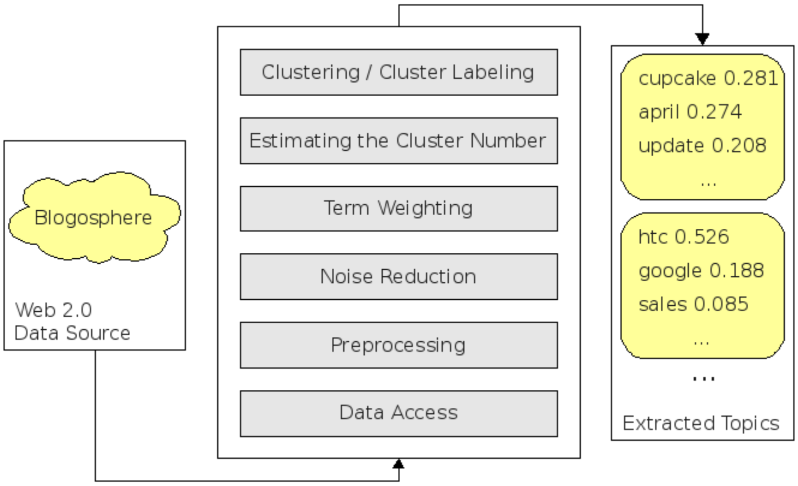


Fig. 2. Topic extraction process steps.

Noise Reduction Very rare and very frequent terms are not considered helpful to characterize articles. As a consequence dimensions representing these terms are removed. To reduce the noise that is inherent in social metadata we experimented with dimensionality reduction based on Latent Semantic Analysis ([9]). However the positive impact of the application of this technique still has to be examined in greater depth.

Term Weighting Terms that appear frequently in the data/metadata of one article but rarely in the whole corpus are likely to be good discriminators and should therefore obtain a higher weight. We use the TF-IDF measure ([10]) which is widely applied in information retrieval systems in order to achieve this goal.

Clustering and Cluster Labeling To be able to cluster the blog articles we need to find a reasonable number of clusters in our data first. For this purpose we follow an approach which is based on the residual sum of squares (RSS) in a clustering result. For document clustering and cluster label extraction we apply non-negative matrix factorization.

We *estimate the number of clusters* in the data set as described in [11], page 365. First we define a range in which we expect to find the number of topics per run. We chose a range between 2 and 20 for our experiments, however the boarders are configurable in our algorithm. For each potential cluster size k ($2 \leq k \leq 20$) we run K-Means i-times (we chose $i = 10$), each time with a different initialization. We compute for each clustering the residual sum of squares (RSS) and the minimum RSS over all i clusterings (denoted by $\widehat{RSS}_{min}(k)$). Then we take a look at the values $\widehat{RSS}_{min}(k)$ and search for the points where successive decreases in $\widehat{RSS}_{min}$ become significantly smaller.[9] The first five such values $k-1$ are stored as reasonable cluster sizes. We store five values in order to enable clusterings according to different granularities. If broad clustering granularity is desired we take the first reasonable number of clusters, for middle granularity the second, and so on.

Using non-negative matrix factorization (NMF) for *document clustering* has firstly been introduced by Xu et al. ([12]). The authors show that NMF-based document clustering is able to surpass latent semantic indexing and spectral clustering based approaches.

NMF finds the positive factorization of a given positive matrix. It is applied on the term-document matrix representation of the document corpus. In the latent semantic space which is derived by applying NMF, each axis represents the base topic of a document cluster. Every document is represented as an additive combination of these base topics. Associating a document with a cluster is done by choosing the base topic (axis) that has the highest projection value with the document. Formally NMF is described as follows:

Let $W = \{f_1, f_2, ..., f_m\}$ be the set of terms in the document corpus after our preprocessing steps. The weighted term vector X_i of a document is defined as

$$X_i = [x_{1i}, x_{2i}, ..., x_{mi}]^T \tag{1}$$

with x_{ij} being the TF-IDF weights of the terms f_i as described before.

We assume that our document corpus consists of k clusters. The goal of NMF is to factorize X into non-negative matrices U $(m \times k)$ and V^T $(k \times n)$ which minimize the following objective function:

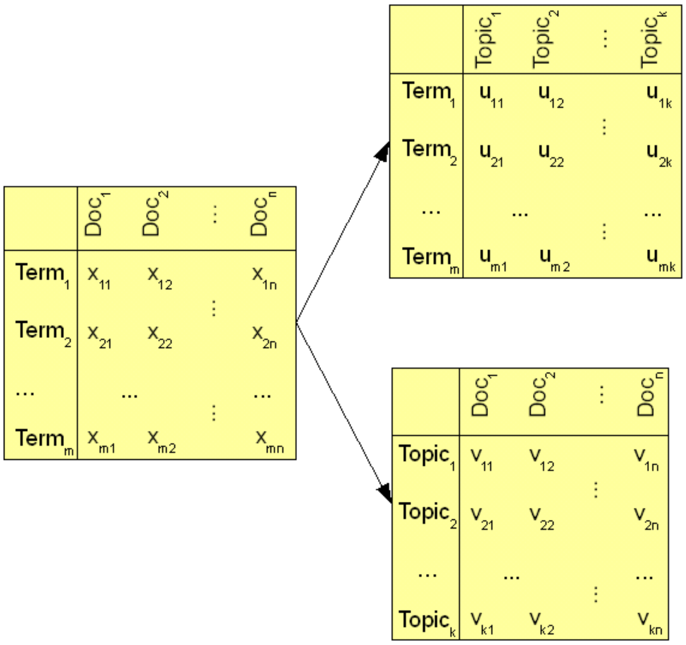
$$J = \frac{1}{2} \parallel X - UV^T \parallel \tag{2}$$

$\parallel \cdot \parallel$ denotes the squared sum of all the elements in the matrix.

Each element u_{ij} of matrix U determines the degree to which the associated term f_i belongs to cluster j. For cluster labeling we simply choose for each cluster the ten terms with the highest degree of affiliation. Analogously each element v_{ij} of matrix V represents the degree to which document i is associated with cluster j. To cluster the documents, again we assign every document to the cluster with the highest degree of affiliation. If a document i clearly belongs to one cluster x then v_{ix} will have a high value compared to the rest of the values in the i'th row vector of V. The matrix factorization is depicted in Figure 3.

In our previous work, we used the X-means algorithm ([13]) to cluster our blog articles. For cluster label extraction, frequency-based cluster labeling as well as feature selection methods such as mutual information and the chi-square test ([11] pages 396-398) have been used. Our experiments showed that the results

[9] $RSS_{min}(k)$ is a monotonically decreasing function in k with minimum 0 for $k = N$ with N being the number of documents.



$$\begin{array}{c|cccc} & Doc_1 & Doc_2 & \cdots & Doc_n \\ \hline Term_1 & x_{11} & x_{12} & & x_{1n} \\ Term_2 & x_{21} & x_{22} & \vdots & x_{2n} \\ \dots & & \dots & & \dots \\ Term_m & x_{m1} & x_{m2} & \vdots & x_{mn} \end{array}$$

$$\begin{array}{c|cccc} & Topic_1 & Topic_2 & \cdots & Topic_k \\ \hline Term_1 & u_{11} & u_{12} & & u_{1k} \\ Term_2 & u_{21} & u_{22} & \vdots & u_{2k} \\ \dots & & \dots & & \dots \\ Term_m & u_{m1} & u_{m2} & \vdots & u_{mk} \end{array}$$

$$\begin{array}{c|cccc} & Doc_1 & Doc_2 & \cdots & Doc_n \\ \hline Topic_1 & v_{11} & v_{12} & & v_{1n} \\ Topic_2 & v_{21} & v_{22} & \vdots & v_{2n} \\ \dots & & \dots & & \dots \\ Topic_k & v_{k1} & v_{k2} & \vdots & v_{kn} \end{array}$$

Fig. 3. Factorization of the term-document matrix by the NMF algorithm.

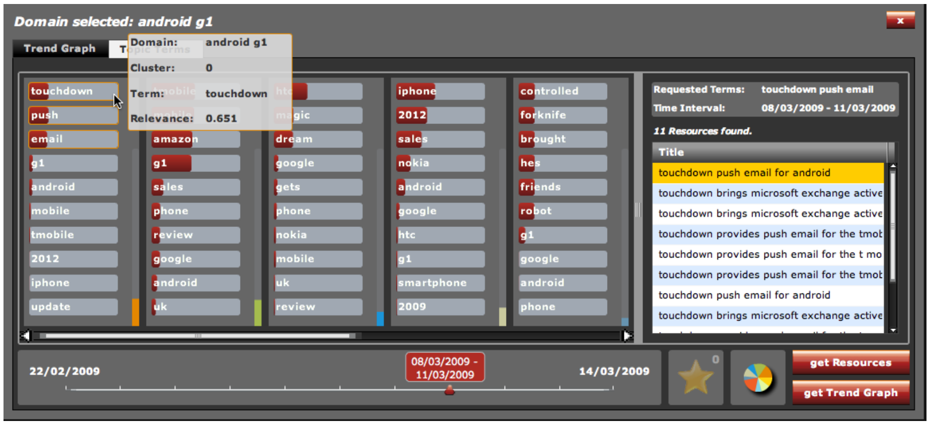
using NMF were more gratifying, in particular with respect to meaningful cluster labels.

6 Information Access

For each run the detected cluster labels are displayed to the user. Every cluster is associated with a label consisting of at most ten terms and their respective relevance values. The presentation of topic terms together with their associated relevance values in the Social Media Miner web interface is shown in Figure 4. Our system offers three options to access the relevant postings of a topic:

1. Get all postings in a cluster.
2. Get postings in a cluster that match specified search terms.
3. Get postings of the current run that match specified search terms.

In order to provide access to all relevant postings in the current run and to avoid the presentation of wrongly clustered postings to the user we chose the third option for our GUI. The approach of deducing a cluster label first and then re-querying the input documents has been proposed in the literature before (e. g., [14]). In our system the user is currently required to select the relevant terms of a topic manually. However we plan to automate this step in the next version of the Social Media Miner.



Domain selected: android g1

Domain: android g1
Cluster: 0
Term: touchdown
Relevance: 0.651

Requested Terms: touchdown push email
Time Interval: 08/03/2009 - 11/03/2009

11 Resources found.

Title
touchdown push email for android
touchdown brings microsoft exchange active
touchdown brings microsoft exchange active
touchdown provides push email for the tmot
touchdown provides push email for the t mo
touchdown provides push email for the tmot
touchdown push email for android
touchdown brings microsoft exchange active

22/02/2009 08/03/2009 - 11/03/2009 14/03/2009

Fig. 4. Social Media Miner interface for the presentation of topics in a selected run.

The result set of this query can be evaluated with precision and recall values [11], whereby the precision value is the critical one. The query should ideally return only articles that belong to the topic of interest, i. e., the precision value should be close to 1.0. Given that the system achieves indeed a high precision, a high recall value is not overly important, since a few good articles, or even only one in certain topics, are enough to get all the facts and information the topic contains. However, from a user's intuitive point of view, it is desirable to obtain a longer list of relevant articles for a very active topic, than for a less active one. This expectation can be met with relatively constant recall values over the queries, if the precision is reliably high at the same time.

We rank the result list according to the authority value of the articles, as described in Section 4. Thus, the resulting reading recommendation gives the user an impression of the popularity and efficient access to the important and relevant articles of the topic, given that precision and recall values fulfill the conditions postulated before.

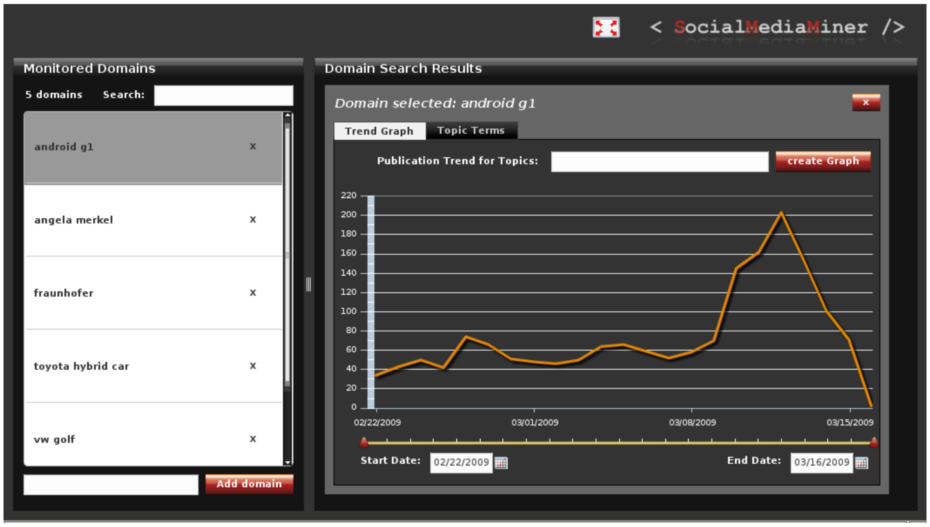
Besides providing access to the relevant blog postings of a topic in a run, we also want to support the users in identifying whether a topic of interest is of increasing or decreasing importance in the blogosphere. For that purpose, the user chooses a topic of interest, again selects the relevant terms of the topic and clicks the "get Trend Graph" button. A graph is generated and displayed that depicts the publication trend of articles matching the selected terms during the time the domain has been tracked. That way the user can easily determine the current relevance of the associated topic in the blogosphere.

7 Evaluation

7.1 Example Data Set

We evaluate our system by comparing its suggestions with manually categorized topics for the articles. As a test domain, we have chosen the relatively new and hyped mobile handset G1 launched by T-Mobile in 2008, with its Google Android software as the most important feature. This is a domain that is interesting for marketing professionals or market researchers in the mobile phone sector.

The resulting search query is "Android G1", and the data has been acquired as described in Section 3 at the end of March 15th 2009, with a time frame of the last 22 days, including February 22nd 2009 as day 0. The search services returned 2193 unique URLs, from which our heuristics validated 1710 as blog articles of the domain, for which a timestamp and the content were available. From the originally 693 links between these articles, only 350 adhered to our criteria, but are supposed to be expressive. The 1710 articles were published in 931 different blogs. Between these blogs, we detected 264 blogroll links.



5 domains

Domain selected: android g1

Start Date: 02/22/2009

End Date: 03/16/2009

Fig. 5. Social Media Miner interface for the publication trend in a domain (i. e., number of blog articles per day).

Figure 5 shows the number of published articles by day in our test data set. This is how a traditional trend analysis would present the data, as outlined in Section 2. Obviously, this method detects three peaks of activity in our data set, leaving the observer alone to find out about the reasons and topics behind them. With almost 400 blog articles in the main peak on the days 17 and 18, this is a time-intensive task to perform for a human being.

7.2 Ground Truth

For the evaluation, we have looked through all of the articles and categorized them into topics. A topic is relating multiple articles by either a specific event in the domain, or by a common entity, which is not the domain itself of course. We found 57 different topics with at least three articles in the data set. 775 blog articles did not belong to any topic, e. g., reviews of an author's new G1 phone.

Figure 6 plots the Top 7 topics of the domain with their volume of articles per day. The ground truth reveals that there exist two different kinds of topics, which are very good to distinguish from each other. Event-based ones and entity-based ones. The articles of an event-based topic usually appear around a certain peak day, in a frame between three and five days, like the announcement of the cupcake update for April. Entity-based topics on the other hand appear more or less intensive throughout the whole time period, e. g., the speculations and discussions about HTC's next android device codenamed "Magic".

Another observation is the composition of peaks in the overall publication trend. Figure 6 disguises that the publication peak at the end of the observed period (see Figure 5) is not expressive for the domain, but mostly a result of four concurrent large-volume events, which occur around the same time by coincidence. This illustrates very well the limits of "Trend Graphs" as they are used in search services today, and the need for a topic detection that can explain the publication trend and its composition in more detail.

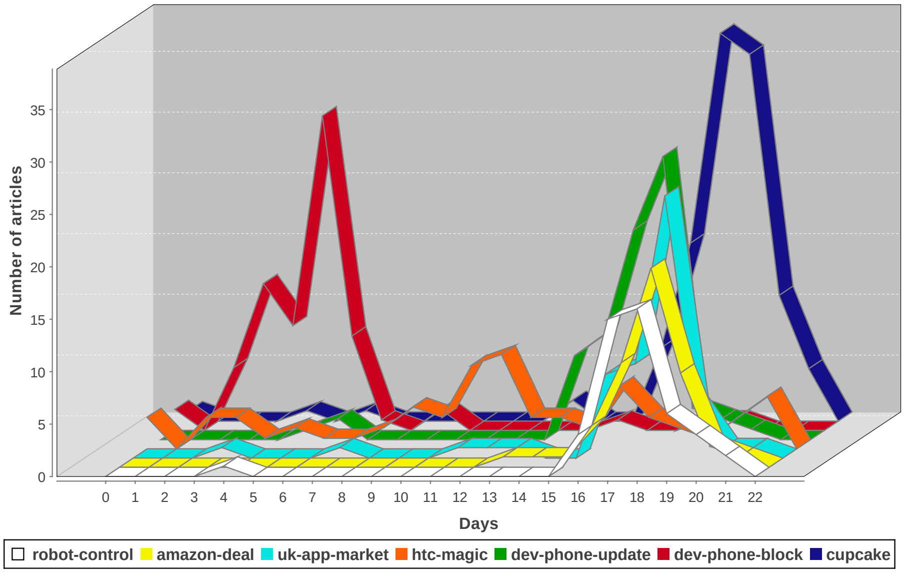


Fig. 6. Ground Truth: Topics in example domain.

7.3 Topic Detection

For the topic detection we aim to identify such topics for which at least 10 blog postings are available in a run. To evaluate the topic detection step, we assigned each cluster whose label indicated a certain topic to the respective topic in our ground truth data set. To objectify this manual step, we require that our system presents at least three relevant articles for the topic in the top ten list of recommended blog postings which corresponds to a precision of 0.3. However the average precision is much higher (> 0.8). Table 1 shows how many topics have been identified in each run. Altogether 29 of 37 topics (78.38%) could be detected with our approach.

Table 1. Detected topics in runs.

Run	#Articles	#Topics to detect	#Topics detected
0 - 3	169	1	1
2 - 5	232	1	1
4 - 7	239	2	2
6 - 9	195	2	2
8 - 11	226	1	1
10 - 13	241	1	1
12 - 15	239	2	1
14 - 17	435	8	8
16 - 19	663	9	6
18 - 21	528	10	6

7.4 Information Relevance

To get access to the relevant postings of a topic the user selects the terms that best characterize the topic and clicks on a search button. We chose three heuristics for the term selection step. Top3 refers to the three most relevant terms in the label, Top5 and Top10 to the five and ten most relevant terms respectively. Additionally we require that the relevance of each selected term is not less than 50% of the relevance of the most relevant term. Terms whose relevance is less are dismissed.

Precision and recall values are calculated over the top 10 recommended blog articles for each topic. With the Top3 heuristic we achieved a precision of 0.84 and with the Top5 and Top10 terms a precision of 0.87 in averaged over all detected topics. The average recall values were 0.35 for the Top3 and Top5 approaches and 0.33 for the Top10 approach. The average precision and recall values for the different term selection approaches are summarized in Table 2. Concerning the recall values it has to be considered that the amount of postings on a topic is usually too high for the user to check all of them. For that purpose we aim at finding a smaller set of relevant postings for each topic. With an average precision of 0.87 for the Top5 and Top10 approaches we can in general present the user eight to nine relevant postings for each topic. In our future work we will examine how the amount of relevant postings can be restricted to a smaller set by exploiting the relevance values derived from the SNA algorithms thus making higher recall values possible.

Table 2. Average precision and recall values for different term selection heuristics.

	Top3	**Top5**	**Top10**
Precision	0.84	0.87	0.87
Recall	0.35	0.35	0.33

8 Conclusion and Future Work

By combining methods from social network analysis and textual data mining, we set up a system for the semi-automatic analysis of topics and trends in selected domains in the blogosphere, therewith supporting the work of professionals in market research or public relations businesses.

In the next version of our system we plan to automate the connection of topic terms with relevant blog postings of the associated topic thus making the manual term selection step obsolete. Further we plan to integrate topic tracking algorithms that allow for a visualization of publication trends of specific topics that way improving the perceptibility of trends in an early stage.

A new insight revealed during this work is the fact that links between blog articles cannot only be used to measure article authority, but they also give strong hints for the topic clustering in the domain. We intend to integrate this network component information into the clustering algorithm and improve it further that way.

Acknowledgments This research has been financed by the IBB Berlin in the project "Social Media Miner", and co-financed by the EFRE fonds of the European Union. Special thanks to Fernanda Pimenta for graphical assistance.

References

1. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. World Wide Web **8**(2) (2005) 159–178
2. Glance, N., Hurst, M., Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: In WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, ACM Press (2004)

3. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. (1998) 194–218
4. Schult, R., Spiliopoulou, M.: Discovering emerging topics in unlabelled text collections. In Manolopoulos, Y., Pokorný, J., Sellis, T.K., eds.: ADBIS. Volume 4152 of Lecture Notes in Computer Science., Springer (2006) 353–366
5. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report;, Stanford University (1998)
6. Obradovic, D., Baumann, S.: Identifying and analysing germany's top blogs. In: Proceedings of the 31st German Conference on AI, Springer (2008) 111–118
7. Wasserman, S., Faust, K., Iacobucci, D.: Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press (1994)
8. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. In: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, AAAI Press (1998) 668–677
9. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41**(6) (1990) 391–407
10. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation **28**(1) (1972) 11–21
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Online edn. Cambridge University Press (April 2009)
12. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM (2003) 267–273
13. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2000) 727–734
14. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on singular value decomposition. In: Intelligent Information Systems. (2004) 359–368