Discovering Hot Topics in the Blogosphere

M. Platakis[1], D. Kotsakos[1], D. Gunopulos[1]

[1]Department of Informatics and Telecommunications, University of Athens

{platakis, d.kotsakos, dg}@di.uoa.gr

Abstract

Over the last few years, blogs (web logs) have gained massive popularity and have become one of the most influential web social media in our times. Every blog post in the blogosphere has a well defined timestamp, which is not taken into account by search engines. By conducting research regarding this feature of the Blogosphere, we could attempt to discover hot topics during a time interval. We present Kleinberg's algorithm about finding bursty and hierarchical streams; we apply it on titles of blog posts to discover bursty terms and compare the results to that produced by Blogscope, an online system for the analysis of the Blogosphere. Finally we present some ideas for future research in the field that could help designing a more sophisticated blog search engine than the existing ones.

Keywords: Blogosphere, blogs, hot topics, burst analysis, text mining.

1. Introduction

Wikipedia defines a blog as a website, usually maintained by an individual, with regular entries of commentary, descriptions of events, or other material such as graphics or video. Over the last few years, blogs (web logs) have gained massive popularity and have become one of the most influential web social media in our times. Anyone with an internet connection can create his own blog for free, using web platforms developed for this specific reason (e.g. blogger.com, wordpress.com etc.). According to blog search engine Technorati there are over 175,000 new blogs every day [Technorati's web site] and over 113 million blogs (not including most non-English blogs) exist today. The huge growth of blogging provides a wealth of information waiting to be extracted.

Blogs define a new area of research concerning information retrieval because blogs' contents have a very specific characteristic not present in traditional web content: a temporal stamp exists in every blog post. Every blog post in the Blogosphere has a well defined value in the temporal axis. Traditional blogs' search engines don't take into account the temporal dimension and treat the blogs as plain web content; at most paying attention to the category tags that usually accompany a post.

By taking into consideration the timestamp of blog posts we can try to detect the period in which the popularity of a specific keyword increases dramatically, thus marking a "burst". We can also attempt to discover interesting (of high burstiness) topics during a specific time period. Further interesting information might be to obtain keyword correlations, bloggers' ranking and influential bloggers and so on.

In this work, we focus on the problem of automatic event detection on blogs. This problem is getting clearly more and more important as the number and size of large time-stamped collections increase (such as blogs). Term burstiness has been extensively researched as a mechanism to address this problem. We investigate and compare novel approaches to model the burstiness of a term and find correlations between bursty terms.

2. Related Work

[Kleinberg (2003)] addresses the problem of extracting meaningful structure from document streams that arrive continuously over time. He attempts to exploit the role of time, mainly, in his personal e-mail by modeling the text stream using an infinite-state automaton, in which bursts appear as state transitions. Rather than using plain frequencies of the occurrences of words, the algorithm employs a probabilistic automaton whose states correspond to the frequencies of individual words. More specifically, state transitions correspond to points in time around which the frequency of the word changes significantly.

Criticizing simplistic methods which only analyze gaps between continuous message arrivals as risky to providing false results, such as identifying lots of short uninteresting bursts and fragmenting true long bursts into shorter ones, Kleinberg defines his goal; to identify bursts only when they show adequate intensity and in a way that permits the burst to maintain its existence through bigger periods of time despite the non-uniform curve of message arrivals.

His automaton consists of a set of states that correspond to the intensity of a keyword appearance, whereas the burst itself is flagged by a, lower to higher, state transition. By assigning costs to state transitions, one can restrain the number of such transitions, therefore eliminating short bursts and making it easier to discover longer ones. The state transition – burstiness connection forms a hierarchical tree structure, with a long burst of low intensity potentially including several bursts of higher intensity inside it.

The most basic bursty model is the automaton with two states (as used in our approach, see section 4 for more), corresponding to "low" and "high". The time gaps between consecutive events are distributed independently according to an exponential distribution whose parameter depends on the current state. One can refer to this parameter as the rate of the keyword arrivals. A higher state incorporates a bigger rate. Between keyword appearances, the automaton may change state with probability p, remaining in current state with probability $1 - p$, independently of previous emissions and state changes. This probability is defined as $p = n^{-\gamma}$ where $n + 1$ is the number of a keyword's appearances (interarrival gaps of the keyword) over the study period of time and γ is a user defined parameter which Kleinberg usually sets to the default value of 1 (as used in our approach, see section 4 for more).

Given an event stream (i.e. a sequence of interarrival time gaps) for a certain keyword, the algorithm seeks to a find the most possible state sequence that is likely to generate that stream. To do so, a Bayes procedure is applied leading to a cost function. Computing a minimum-cost state sequence is accomplished by dynamic programming techniques. Kleinberg defines a measure of weight associated with each burst and solves the problem of enumerating all the bursts by order of weight. In the two-state automaton, the weight is equal to the improvement in cost incurred by using state q_1 over the interval rather than state q_0.

Concluding, this algorithm generates a ranked list of the most significant word bursts in the document stream, together with the intervals of time in which they occurred. This can serve as a means of identifying topics or concepts that rose to prominence over the course of the stream, were discussed actively for a period of time, and afterwards receded.

[Bansal et al. (2007)] present Blogscope (www.blogscope.net), an online system for the analysis of the Blogosphere that identifies keyword bursts. The analysis is performed on the body of blogposts, and the system does not make use of any tags

that may have been inserted by the bloggers, with this decision being based on a couple of reasons. *Burst* is defined as "unexpected popularity of a keyword within a temporal window". Bursts are divided in two distinct categories: anticipated and surprising, with the main difference between them being the way the popularity of the term increases. Kleinberg's infinite state automaton is judged as interesting but computationally too expensive to be adopted by an online system like Blogscope that needs to compute keyword bursts on the fly. Instead, Blogscope employs an algorithm inspired by [Fung et al. (2005)] who propose techniques to identify sets of bursty features given a text stream. A base popularity μ and variance σ^2 are computed using popularity values of a term for the last w days. Blogscope in its current implementation takes into account popularity values for the last 90 days. Then, using a standard normal curve, outlier days are found and are labeled as burst days for the keyword examined. For this purpose, scores of some scoring functions are aggregated and a list of hot keywords is produced. This statistical approach overcomes difficulties raised by the subjective nature of the notion of "interestingness".

Blogscope introduces a technique to identify keyword correlations that makes use of the notion of popularity curves. The popularity curve of a keyword is a function to time that shows how often this specific keyword appears in the Blogosphere. So, it is expected that related keywords will have similar popularity curves, because they appear with high probability in the same blogposts. Comparing the curves of two keywords would give an insight in their temporal relationship. Another way to quantify the correlation of two keywords is to compute the probability of finding the one in a document that contains the other. Because of the large computational cost that is required to check each pair of tokens to find this probability, Blogscope adopts a fast technique that is similar to the above presented. The tf–idf weight (term frequency–inverse document frequency), often used in information retrieval and text mining, is applied to compute a score that represents the correlation between pairs of terms. Term frequencies, inverse document frequencies and other indices are maintained in separate data structures for efficiency purposes.

Finally, Blogscope adopts some techniques for data preprocessing. Among other tasks, it attempts to remove spam blogposts from input data before its analysis engine is triggered. In order to achieve this, it uses a Bayesian classifier and some simple, empirically formed, heuristics.

[Wang et al. (2007)] propose a probability mixture model, using a statistic distribution of words to describe a topic, for the mining of correlated bursty topics patterns in different text streams (even if the streams have different vocabularies) which arrive simultaneously (i.e. aligned in time). [Agarwal (2008)] presents a model of identifying influential bloggers using several heuristics and parameters such as comments, outlinks, inlinks, posts' length.

Relevant work has been done in the field of finding similarities between time-series [Vlachos et al. (2004)], opinion mining [Ding et al. (2008)], concerning burst detection [Zhang et al. (2006)], bloggers' interests [Cheng et al. (2008)] and further analysis of the Blogosphere [Kumar et al. (2005)], [Chi et al. (2007)].

3. Our Approach

We, initially, attempt to discover bursty terms from blogs' data, i.e. short periods of times in which appearances of specific words increase radically in comparison to the long period we study. We, afterwards, evaluate the accuracy of the bursty terms, by trying to match them with real life events that took place in the bursty period of time.

A certain story is formed by a group of many correlated terms. As the popularity of as specific topic diminishes, this group ceases to exist. We try to obtain keywords correlations, in order to automatically identify such groups. We address this problem, assuming that related keywords produce similar activity as far as burstiness is concerned and therefore comparing their weight and period of their bursty curve (i.e. curve plotting their burstiness during the period of study). We evaluate these correlations by trying to mine a real life topic that could have formed such a group of keywords. Last but not least, we identify this topic with a real life event that took place in the period of study.

3.1 Pre-processing

In order to remove noise from data, the following pre-processing steps are performed:

- Using a dictionary based script including 628 words; we remove several stop-words, like common verbs or adverbs. The reason of doing this is that such words, although being bursty sometimes, cannot represent hot topics. We do not remove numbers, because they may correspond to (e.g. an important date or age), about which bloggers may have written many times in a short time interval.

- The next step is to convert all letters to lowercase, so that words like "This" and "this" are considered the same.

We did not use stemming techniques because most of them retain only a very basic root of each word, making it difficult to reason on the results produced by the algorithm. For example, words *fitness* and *fit* would both be converted to *fit*, quite changing the semantics of the word as far as the detection of hot topics is concerned.

3.2 Kleinberg's Automaton on Blogs

Our decision to apply Kleinberg's automaton on blog data was based on the algorithm's notion of temporal bursts in a sequence of documents combined with Blogscope's team's skepticism of adopting it. We performed our experiments on the titles of blog posts (as opposed to Blogscope's usage of the whole body of each blog post) in order to determine if they are suffice to mine the underlying bursts as well as to gain some computational time. Titles had to be parsed through the permalinks of the blog posts (i.e. the permanent url of a certain blog post), which could sometimes lead to incomplete or obscure title extraction.

We used a two-state automaton to conduct our experiments. Gamma parameter, as described in section 2, is set to 1, the rate of q_0 is set to n/T (n as described in section 2 and T is the length of the period of time we study). The ratio of rate of q_1 to rate of q_0 is set to 2 (i.e. double rate for the "bursty state"). The titles are time stamped from 1 to 10, corresponding to the day each blog post appeared, during our 10-day period of study.

4. Experimental results

4.1 Bursty Keywords Examples and Comparison with Blogscope

Below, we present some keyword examples that came up as bursty, after Kleinberg's automaton-based method was applied on blog posts' titles, try to connect them with "real life" facts (if such exist) that would elaborate on the burstiness of these keywords as well as if these specific keywords appear as hot ones on Blogscope for the first ten days of April 2008. Kleinberg's algorithm's output displays only the bursty keywords of the dataset in the following form:

"Keyword, weight, start_day, end_day"

The term *fool* appears in our results with the record "fool,17.343,1,2" which means that *fool* was recognized as a bursty term for the first two days of April 2008 with a weight of 17.343. When the output is sorted by weight in descending order, *fool* is ranked 25th. The appearance of *fool* in the bursty keywords, for the first and the second day of April, is justified by the bloggers' intention to blog these days about April's Fools' Day. When checked on Blogscope for the same time period, *fool* produced the popularity curve depicted on Figure 1, thus providing similar results to our approach for the 1st and the 2nd day of April 2008.
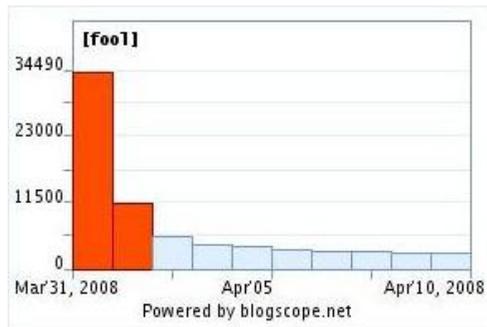

Figure 1. *Popularity curve for the term "fool" for the period 1-10 April 2008 produced by Blogscope*

The term *preprocessor* appears in Kleinberg's automaton's results with the record "preprocessor,85.947,6,6" which means that *fool* was recognized as a bursty term for April 6th 2008 with a weight of 85.947. When the output is sorted by weight in descending order, *preprocessor* is ranked 3rd. After a Blogscope query for the same time period, *preprocessor* produced the curve depicted on Figure 2, showing that it was a hot keyword for April 7th 2008. A little difference between Kleinberg's algorithm's output and Blogscope's results shows up. Demonstrated in the following examples, this one-day difference between the two methods' results appears quite often, with some bursty keywords appearing on our results one day earlier than they appear on Blogscope's curves.

Examined more carefully, the burstiness of the term *preprocessor* on April 7th can be justified by the large amount of identical spam blog posts that were uploaded that day and contained the phrase "All C/C++ compilers implement a stage of compilation known as the preprocessor" among others. Neither Blogscope nor Kleinberg's algorithm managed to recognize these blog posts as spam, and both labeled the term *preprocessor* as hot or bursty for that day.
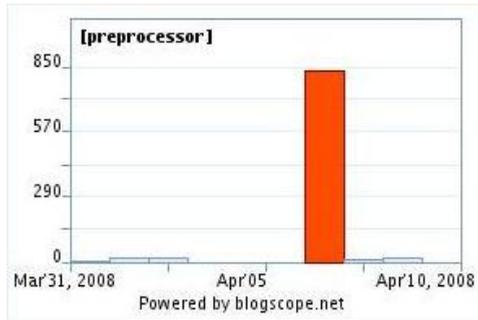
Figure 2. *Popularity curve for the term "preprocessor" for the period 1-10 April 2008 produced by Blogscope*

One last example of a keyword that is recognized as bursty both by Blogscope and the automaton is *mp3*. The term *mp3* appears in Kleinberg's algorithm's output with the record "mp3,78.431,6,6" which means that *mp3* was recognized as a bursty term for April 6th 2008 with a weight of 78.431. When the output is sorted by weight in descending order, *mp3* is ranked 6th. When checked on Blogscope for the same time period, *mp3* produced the curve depicted on Figure 3, showing that it was a hot keyword for April 6th 2008.
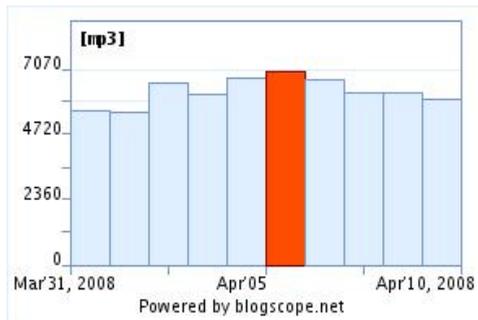

Figure 3. *Popularity curve for the term "mp3" for the period 1-10 April 2008 produced by Blogscope*

4.2 Finding Clusters of Correlated Keywords

Keywords in the Blogosphere may be correlated, and one way to find clusters of correlated keywords is to group them according to the similarity of their popularity curves, which is a time series analysis' problem. In Blogscope users can specify a

time interval for which a list of keywords correlated to query keywords is produced. While processing Kleinberg's automaton's results, we followed the assumption to label as *correlated*, keywords that appear to be bursty for the same time interval, or at least investigate if they really are. For example keywords *credit* and *card*, which appear on our results with the records "credit,115.075,2,2" and "card,42.598,2,2" respectively are assumed to be correlated. This assumption has the disadvantage that it does not take into account keywords that are not both bursty, because the results only contain the bursty ones. Regarding the example given above, Blogscope produces the list of correlations depicted on Figure 4, when queried with the term *credit* for April 3rd 2008 (taking into account the one-day difference between the automaton's and the Blogscope's results).
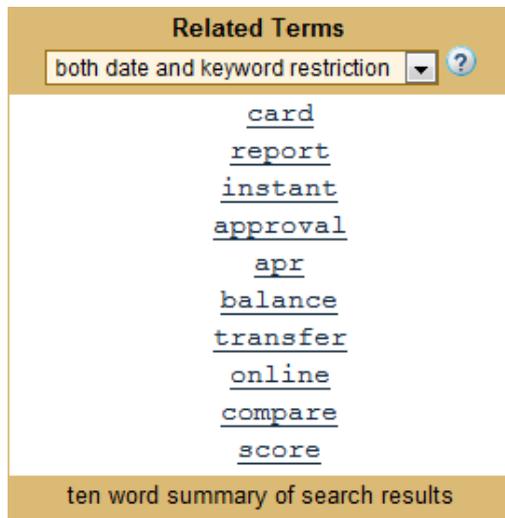

Related Terms

both date and keyword restriction

card
report
instant
approval
apr
balance
transfer
online
compare
score

ten word summary of search results

Figure 4. List of correlations to term "credit" for April 3rd 2008 produced by Blogscope

Important to mention is the fact that the depicted list is sorted by quantified correlation between *credit* and the corresponding terms in descending order. Table 1 shows a list that contains the records produced by Kleinberg's automaton and correspond to the records showed in Figure 4. This list is sorted by weight in descending order. It appears that the order in which the terms appear in both lists is almost the same. The time interval in which all the terms appear to be bursty is April's 2nd, which means that the assumption regarding the extraction of keyword correlations form Kleinberg's algorithm's output made above works quite well.

Table 1. *List of correlated keywords to term "credit" according to Blogscope, as seen in Kleinberg's algorithm's results.*

credit,115.075,2,2

card,42.598,2,2

report,14.336,2,2

instant,11.769,2,2

reports,9.89,2,2

balance,7.607,2,2

apr,7.065,2,2

transfer,4.441,2,2

Another example of finding bursty keyword correlations using Kleinberg's algorithm, and extracting them according to the time interval in which the keywords are bursty is the cluster formed around the keyword *heston.* The fact that justifies the burstiness and the cluster that follows is that "Charlton Heston was an actor, who died on Saturday, April 5, 2008 at the age of 84". Searching through Kleinberg's algorithm's output for the terms {charlton, heston, actor, dead, 84, actor} produced the list of Table 2, sorted by descending weight. All of these terms appear to be bursty for April 5th, 2008, which is the day when Heston died, and bloggers wrote about his death. A similar list of correlated keywords is produced by Blogscope on April 6th, a day after Heston's death.

Table 2. *List of keywords {charlton, heston, actor, dead, 84, actor} as seen in Kleinberg's algorithm's results.*

heston,12.213,5,5
charlton,9.929,5,5
actor,4.302,4,5
dead,3.388,5,5
84,3.08,5,5

5. Future Work

We plan to continue our research on the field. We would like to attempt to improve the performance of our experiments by tweaking the automaton's parameters, employing other algorithms as well, being able to submit a query containing more than one word as a keyword and probably taking into consideration multimedia files often posted on blogs etc.

We will try experimenting on the whole bodies of the blog posts but we would like to point out that usage of titles in aggregation with tagging in blogs (ignored by the Blogscope system) may be able to produce interesting results and we plan to investigate on this assumption. Moreover, we desire to develop an automated method to discover keyword correlations, using an index to speed up the process, extracting a popularity curve for each keyword from the automaton's output and employing an appropriate distance metric in order to compare such curves.

Last but not least, we plan to extend our study field by performing research on famous micro blogging services (such as Twitter.com) and news' services which provide far more continuous temporal text streams. Our initial approach will be the transition from daily time stamps to hourly time stamps, which will provide us with a perception of what "happens to the world" during each day.

6. Acknowledgements

7. References

1. Agarwal N. (2008), *Identifying the influential bloggers*, ACM WSDM conference.
2. Bansal N. and Koudas N. (2007), *Searching the Blogosphere*, WebDB Workshop.
3. Cheng Y. and Qiu G. (2008), *Model bloggers' interests based on forgetting mechanism*, International World Wide Web Conference (WWW).
4. Chi Y., Zhu S., Song X., Tatemura J. and Tseng B. (2007), *Structural and temporal analysis of the blogosphere through community factorization*, ACM SIGKDD Conference (KDD).
5. Ding X., Liu B. and Yu P. (2008), *A holistic lexicon-based approach to opinion mining*, ACM WSDM conference.
6. Fung G.P.C., Yu J.X., Yu P.S. and Lu H. (2005), *Parameter free bursty events detection in text streams*, VLDB, pages 181-192.
7. Kleinberg J. M. (2003), *Bursty and hierarchical structure in streams*, Data mining and Knowledge Discovery, 7(4): 373-397.
8. Kumar R., Novak J., Raghavan P. and Tomkins A. (2005), *On the bursty evolution of the blogspace*, International World Wide Web Conference (WWW).
9. Technorati's website: http://technorati.com/about/
10. Vlachos M., Meek C., Vagena Z., Gunopulos D. (2004): *Identifying Similarities, Periodicities and Bursts for Online Search Queries*. SIGMOD Conference: 131-142
11. Wang X., Zhai C., Hu X. and Sproat R. (2007), *Mining correlated bursty topic patterns from coordinated text streams*, ACM SIGKDD conference (KDD).
12. Zhang X. and Shasha D. (2006), *Better burst detection*, IEEE International Conference on Data Engineering (ICDE).