

# Detecting Buzz from Time-Sequenced Document Streams

Jeonghee Yi  
IBM Almaden Research Center  
650 Harry Rd.  
San Jose, CA 95120  
jeonghee@almaden.ibm.com

## Abstract

*This paper presents a formal method of detecting emerging and changing interests that appear in document streams arriving continuously over time. Examples of such document streams include email, news articles, and weblogs (or blogs). We utilize the temporal information associated with documents in the streams and discover emerging issues and topics of interest and their change by detecting buzzwords in the documents. Buzzwords are terms that occur with strong momentum for a relatively short period of time.*

*Our approach for buzz detection is based on the notion of “burst of activities” proposed by Kleinberg [7]. The burst of activities is modeled using a weighted automaton. We propose an algorithm to discover buzzwords of high intensity measured by their momentum and relative duration of the bursts. The method is applied and validated on a stream of blog postings and we report the experiment results.*

## 1. Introduction

The goal of the present work is to develop a formal method of modeling and computing emerging and changing interests appearing in document streams arriving continuously over time. As a way to see the emergence of new topics of interests or the change of interests, we detect buzzwords in a set of documents over time. Buzzwords are terms that occur with high *momentum* for a relatively short period of time preceded or followed by longer pauses in close proximity. By analyzing the sequence of episodes reflected in the buzzwords, we can observe when new topics are arising and how the concentration of interests is shifting over time.

We base our buzz detection on the burst event model recently developed by Kleinberg [7]. He models bursty events using an infinite-state automaton that emits messages at different rates depending on its state. A set of states in the automaton corresponds to increasingly rapid rates of emis-

sion, and the onset of a burst is signaled by a state transition from a lower state to a higher state. By assigning costs to state transitions, one can control the frequency of such transitions, preventing very short bursts and making it easier to identify the lasting periods of bursts despite transient changes in the rate of the stream.

Kleinberg’s burst event model enables us to find low cost state transition sequences and bursty events generated by states that emit messages at high rate. We adopt this bursty event model and propose an algorithm that detects high intensity buzz (or buzzwords) from bursty events. Since the model does not take into account the duration of the bursty events or provide a way to measure the weights of bursty events on in a continuous stream, the model is not directly applicable to detect buzzwords. For our definition of buzzwords, not all bursty events qualify to be buzzwords.

Our buzzword detection algorithm takes into account the relative duration and the momentum of bursty events to measure the degree of concentration of bursty events. In short, we define buzzwords as terms with bursty activities of high concentration and show that they can be found by computing relatively short bursty events with high momentum.

The buzz detection can be highly useful for many applications including market and consumer trend analysis, politics, and intelligence. For example, by capturing the change of consumer interests, or growing interests (or concerns) on certain aspects of their products among various groups of target consumers, companies can better respond to such interests and concerns, and establish more effective marketing strategies.

The rest of this paper is organized as follows: we first review the model of burst streams. In Section 3, we propose the formal definition of buzzwords and describe an algorithm of detecting them. Then we describe our empirical study on blog postings and report experimental results of buzzword detection on political blog postings in Section 4. Related works are reviewed in Section 5. Finally, we conclude with discussion.

## 2. Background

### 2.1. Burst Analysis

Kleinberg’s work on identifying bursts in a stream of events [7] is reviewed in this section. For document streams that arrive continuously over time, he observed that the appearance of a topic in a document stream is signaled by a “burst of activity,” with certain features rising sharply in frequency as the topic emerges. For instance, an event might correspond to the appearance of an email containing particular keywords.

The approach is based on modeling the stream using an infinite-state automaton, in which bursts appear naturally as state transitions. The bursts associated with state transitions form a naturally nested structure, with a long burst of low intensity potentially containing several bursts of higher intensity inside it (and so on, recursively). This can provide a hierarchical decomposition of the temporal order, with long-running episodes intensifying into briefer ones according to a natural tree structure. There is a cost associated with any state transition to discourage short bursts. Given an event stream the method finds a low cost state sequence that is likely to generate that stream. Finding an optimal solution to this problem can be accomplished by dynamic programming.

### 2.2. A Generative Model for Two-State Automaton

In this section, we describe the generative model to find a likely sequence of state transition from a set of time-stamped pages. Since we aim at identifying bursts of high intensity, but not to emphasize the hierarchical structure of the bursts, we adopt the *two-state* automaton [7], where the generation of events by the automaton is modeled with two states, “low” and “high.” In the high state events are generated at rapid rate, thus to form bursts of events, and at slow rate in the low state. The time gaps between consecutive events are distributed independently according to an exponential distribution whose parameter depends on the state. Messages are emitted in a probabilistic manner, so that the gap  $x$  in time between messages  $i$  and  $i + 1$  is distributed according to the exponential density function  $f(x) = \alpha e^{-\alpha x}$ , where  $\alpha > 0$ . That is, the probability that the gap exceeds  $x$  is equal to  $e^{-\alpha x}$ .  $\alpha$  is the rate of message arrivals. The expected value of the gap in this model is  $e^{-1}$ .

The burst event model extends this simple formulation by exhibiting periods of lower rate interleaved with period of higher rate. Let  $q_0$  and  $q_1$  correspond to the two states, “low” and “high”, of the automaton. When  $A$  is in state  $q_0$ , messages are emitted at a slow rate, with gaps  $x$  between consecutive messages distributed independently according to a density function  $f_0(x) = \alpha_0 e^{-\alpha_0 x}$ . When  $A$  is

in state  $q_1$ , messages are emitted at a faster rate, with gaps distributed independently according to  $f_1(x) = \alpha_1 e^{-\alpha_1 x}$ , where  $\alpha_1 > \alpha_0$ . Finally, between messages,  $A$  changes state with probability  $p \in (0,1)$ , remaining in its current state with probability  $1-p$ , independently of previous emissions and state changes.

With the model, a sequence of messages is generated as follows.  $A$  begins in state  $q_0$ . Before each message is emitted,  $A$  changes state with probability  $p$ . A message is then emitted, and the gap in time until the next message is determined by the distribution associated with  $A$ ’s current state.

### 2.3. Finding Optimal State Sequences

The generative model can be applied to find a likely state sequence, given a set of messages. Suppose there is a given set of  $n+1$  messages, with specified arrival times. They determine a sequence of  $n$  inter-arrival gaps  $X = (x_1, x_2, \dots, x_n)$ , where  $x_i > 0$ . Let  $Q = (q_{i_1}, q_{i_2}, \dots, q_{i_n})$  be a state sequence. Each state sequence  $Q$  induces a density function  $f_Q$  over sequences of gaps, which has the form  $f_Q(x_1, x_2, \dots, x_n) = \prod_{t=1}^n f_{i_t}(x_t)$ . If  $b$  denotes the number of state transitions in the sequence  $Q$  – that is, the number of indices  $i_t$  where  $q_{i_t} \neq q_{i_{t+1}}$  – then the prior probability of  $Q$  is equal to

$$\begin{aligned} \left( \prod_{i_t \neq i_{t+1}} p \right) \left( \prod_{i_t = i_{t+1}} 1 - p \right) &= p^b (1 - p)^{n-b} \\ &= \left( \frac{p}{1-p} \right)^b (1-p)^n \end{aligned}$$

Let  $i_0 = 0$ , since  $A$  starts in state  $q_0$ . Then, the conditional probability of a state sequence  $Q$  given inter-arrival gaps  $X$  is

$$\begin{aligned} Pr[Q | X] &= \frac{Pr[Q] f_Q(X)}{\sum_{Q'} Pr[Q'] f_{Q'}(X)} \\ &= \frac{1}{Z} \left( \frac{p}{1-p} \right)^b (1-p)^n \prod_{t=1}^n f_{i_t}(x_t) \end{aligned}$$

where  $Z$  is the normalizing constant  $\sum_{Q'} Pr[Q'] f_{Q'}(X)$ . Finding a state sequence  $Q$  maximizing this probability is equivalent to finding one that minimizes

$$\begin{aligned} -\ln Pr[Q | X] &= b \ln \left( \frac{1-p}{p} \right) + \left( \sum_{t=1}^n -\ln f_{i_t}(x_t) \right) \\ &\quad - n \ln(1-p) + \ln Z \end{aligned}$$

Since the third and fourth terms are independent of the state sequence, the problem is equivalent to finding a state sequence  $Q$  that minimizes the following *cost function*:

$$c(Q | X) = b \cdot \ln \left( \frac{1-p}{p} \right) + \left( \sum_{t=1}^n -\ln f_{i_t}(x_t) \right)$$

The first of the two terms in the expression favors sequences with a small number of state transitions, while the second term favors state sequences that conform well to the sequence  $X$  of inter-arrival gap values. Thus, one expects the optimum to track the global structure of bursts in the gap sequence, while holding to a single state through local periods of non-uniformity.

### 3. Buzzword Detection

In Section 2.3, we discussed a method that computes an optimal state transition sequence of a given set of inter-arrival times of events. For the given optimal state sequence, [7] defines bursty events as events generated at high arrival rate by high states ( $q_1$ ). We use the bursty events for the detection of buzzwords. Buzzwords are terms of high momentum for a relatively short period of time.

Note that not all bursty events by the Kleinberg's model can be considered as buzz because the model does not take into account the relative duration and the mass of the bursts. We measure the degree of concentration of bursty events in terms of relative duration and momentum. We show that terms with bursty activities of high degree of concentration qualify to be buzzwords for the corresponding time periods. When a term appears in a document stream with bursty pattern, we consider it as a potential candidate for buzzwords and compute the degree of concentration of the candidate. If the degree of concentration is high enough (i.e., higher than a given threshold), the corresponding term is considered to be a buzzword.

Suppose there is a given set of  $n+1$  messages,  $D = (d_0, d_1, \dots, d_n)$ , with specified arrival times. They determine a sequence of  $n$  inter-arrival gaps  $X = (x_1, x_2, \dots, x_n)$ , where  $x_i > 0$ . Let  $I_i^+(w)$  be an interval,  $[t_{begin}(I_i^+(w)), t_{end}(I_i^+(w))]$ , of high states of a term  $w$ , and call it a "high-interval of  $w$ ", where  $t_{begin}(I_i^+(w))$  and  $t_{end}(I_i^+(w))$  denote the beginning and the ending time, respectively, of the interval. Likewise, let  $I_i^-(w)$ , a "low-interval of  $w$ ", be the interval of low states of  $w$  immediately after  $I_i^+(w)$ . Let *relevant documents* to be documents containing  $w$  in them. The followings are the notations we use in the formal definition of buzzwords:

- *duration* of  $I_i^+(w)$ :

$$|I_i^+(w)| = t_{end}(I_i^+(w)) - t_{begin}(I_i^+(w))$$

where  $t_{begin}(I_i^+(w))$  and  $t_{end}(I_i^+(w))$  are the beginning and ending time of  $I_i^+(w)$ , respectively.

- *mass* of  $I_i^+(w)$ ,  $m_{I_i^+(w)}$ , is the number of documents containing  $w$  arrived during  $I_i^+(w)$ .

- *arrival rate* of  $I_i^+(w)$ ,  $\lambda_{I_i^+(w)}$ , is the arrival rate of relevant documents during  $I_i^+(w)$ :

$$\lambda_{I_i^+(w)} = \frac{1}{E[X_{I_i^+(w)}]}$$

$X_{I_i^+(w)} = (x_r, x_{r+1}, \dots, x_{s-1})$  is a sequence of inter-arrival gaps of relevant documents,  $D_{I_i^+(w)} = (d_r, d_{r+1}, \dots, d_s)$ , within  $I_i^+(w)$ , where the arrival time of  $d_j$  ( $r \leq j \leq s$ ) belongs to the interval  $[t_{begin}(I_i^+(w)) \leq j \leq t_{end}(I_i^+(w))]$ .

- *span ratio* of  $I_i^+(w)$ ,  $\rho_{I_i^+(w)}$ , is the ratio of the duration of the surrounding low intervals to the duration of the high interval  $I_i^+(w)$ :

$$\rho_{I_i^+(w)} = \frac{|I_{i-1}^-(w)| + |I_i^-(w)|}{|I_i^+(w)|}$$

*Span ratio* is the measure of relative shortness of the duration of a buzzword.

- *momentum* of  $I_i^+(w)$ , or  $\xi_{I_i^+(w)}$ :

$$\xi_{I_i^+(w)} = \lambda_{I_i^+(w)} \cdot m_{I_i^+(w)}$$

- The *concentration* of events in  $I_i^+(w)$ ,  $\kappa_{I_i^+(w)}$ , is defined as follows:

$$\kappa_{I_i^+(w)} = \rho_{I_i^+(w)} \cdot \xi_{I_i^+(w)} \quad (1)$$

Intuitively,  $\kappa_{I_i^+(w)}$  has the following properties:

- the concentration becomes higher with increasing momentum of the interval, either by having higher mass or higher arrival rate.
- the concentration becomes higher with shorter time interval with respect to the lengths of surrounding low intervals.

Given the property,  $\kappa_{I_i^+(w)}$  is a good measure of *buzzness* of events during  $I_i^+(w)$ . Thus, we consider  $w$  qualifies to be a buzzword for time period  $I_i^+(w)$ , if the degree of concentration  $\kappa_{I_i^+(w)}$  is higher than the given threshold  $\phi$ :

$$\kappa_{I_i^+(w)} \geq \phi$$

The following describes the algorithm that determines buzzwords from a document stream:

- For each term  $w$  in the stream, compute the optimal state sequence  $Q(w)$  as described in Section 2.3.
- For each high interval of the state sequence  $Q(w)$ ,  $I_i^+(w)$ , compute the degree of concentration,  $\kappa_{I_i^+(w)}$ , as described in equation (1) in Section 3.
- $w$  is a buzzword if the following holds:

$$\kappa_{I_i^+(w)} = \rho_{I_i^+(w)} \cdot \xi_{I_i^+(w)} \geq \phi$$

## 4. Empirical Study

For the proof of concept, we applied the buzzword detection algorithm on a set of blog messages posted in the first quarter of 2004. Blogs differ from traditional web pages structurally: blogs represent concatenation of messages authored by a single individual. The topics or contents of individual postings of a blog can be highly diverse, unlike a newsgroup posting and the following threads that are typically about the same topic and written by multiple authors. Each blog posting has a highly reliable timestamp typically entered by a blog software at the time of posting.

### 4.1. Data Preprocessing

**Template Removal:** Blogs typically have templates containing profile information of the authors and links to other sites that the authors frequently visit, as well as site specific template. We remove these template portions out of the page content in order to prevent this persistent information from being included into the actual journal content. We applied a template removal algorithm based on the method proposed by [2].

**Page Segmentation:** A natural unit of segmentation for blogs is each posting that is typically separated by the time entry associated with each posting representing the time when the posting is entered. We applied a page segmentation algorithm proposed by [9]. The algorithm identifies blog-specific date pattern and segments at the location of new date entries. It is based on the observation that most blogs are published using a blog publishing software package, and therefore dates are written in a uniform format.

**Detagging HTML Tags** We remove scripts, HTML tags and associated attributes from the resulting blog segments, except a handful of selected attributes: ALT attribute in IMG APPLET, and AREA tags, ABBR in TH and TD, SUMMARY in TABLE, and PROMPT in ISINDEX. We skip the text in between SCRIPT, STYLE, COMMENT, and DEL tags.

**Topic Page Detection** The underlying classifier we used for topic page detection is a variant of the Winnow classifier [10, 11]. For the classifier, we model documents using the standard binary bag-of-words representation, thus disregarding the order and term frequency within documents. The classifier computes the following:

$$h(x) = \sum_{w \in V} f_w \cdot c_w(x)$$

where  $c_w(x) = 1$  if term  $w$  occurs in document  $x$  and  $c_w(x) = 0$  otherwise.  $f_w$  is the weight of term  $w$ . If

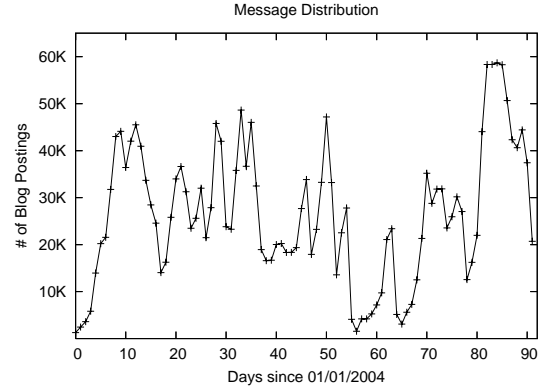


Figure 1. Number of blog postings by day

$h(x) > V$  then the classifier predicts topical, and otherwise predict irrelevant.

### 4.2. The Dataset

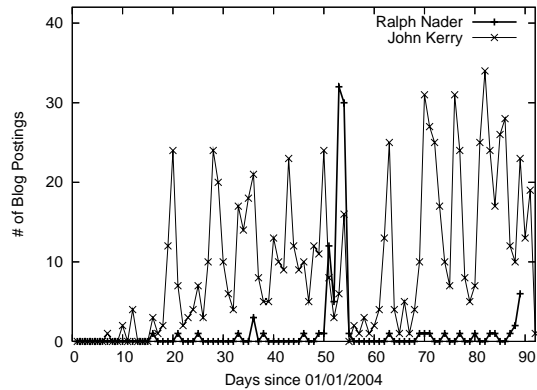
From a general web crawl by WebFountain Crawlers [4], we identified about 1M blog pages. After applying data preprocessing techniques described in Section 4.1, we collected about 29K blog postings with political contents posted in the first quarter of 2004. Figure 1 charts the profile of the blog postings of every day during the period of our study.

### 4.3. Experimental Results

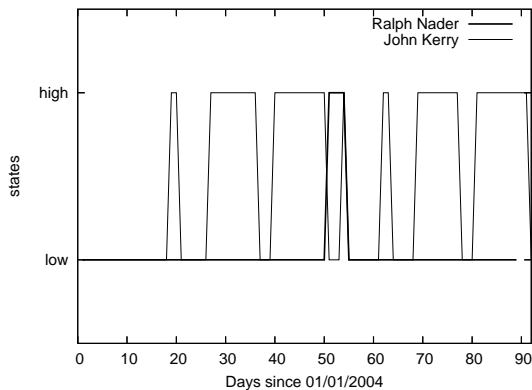
We ran the buzz detection algorithm on the test dataset. For each word in the dataset, we computed an optimal state sequence  $Q(w)$ . Out of  $Q(w)$ , we computed all  $I_i^+(w)$ 's. For each  $I_i^+(w)$ , we computed the degree of concentration (or *buzzness*) and selected ones with high degree of concentration. Additionally, we applied the same algorithm to the bi- and tri-grams of noun phrases identified by parts-of-speech (POS) tagging in order to capture the buzz of like “George W. Bush”, “Iowa Caucus”, or “gay marriage”.

Top 30 buzz words in the decreasing order of their *buzzness* are given in Table 1. Table 2 lists the buzzwords in time sequence showing the change of interests of bloggers over time. The results show that the buzz detection algorithm is able to detect highly bursty terms, while excluding frequent terms that appear quite persistently.

Figure 2 shows the message distribution of “John Kerry” and “Ralph Nader”, two of the highly ranked buzzwords during the experiment period, measured by the number of blog postings containing the terms. Figure 2 illustrates the corresponding state transitions of the terms. The buzz about “Ralph Nader” is quite straightforward as shown in Figure 2



**Figure 2. Buzz on “Ralph Nader” and “John Kerry”**



**Figure 3. State transitions of terms “Ralph Nader” and “John Kerry”**

and captured by the state transition graph. On Feb. 20, 2004, it was announced that Ralph Nader would enter the 2004 presidential race. Later, the formal announcement was made on Feb. 22. Reflecting the news, there was burst of postings about him from Feb. 20th to 23rd, while his name was mostly infrequent for the rest of the period. Our algorithm identified this single buzz which was highly ranked. (See Table 1.)

Of interesting contrast, the occurrences of John Kerry’s name in the dataset show a sequence of bursts of activities. Overall, his name was persistently active for most part of the experimental period since late January. As the result, our algorithm identified only the first burst as buzz and ignored the rest. With persistently high activities, the subsequent bursts got low score of span ratio. This phenomenon brings up an interesting issue: with a little investigation, it is observed that each burst of John Kerry has some events associated such as his victory of Iowa Caucus, and his victory on Super Tuesday. Our algorithm was able to detect buzz on the individual events (such as “Super Tuesday” or “Iowa

| Rank | Buzzwords      | Duration (days) |
|------|----------------|-----------------|
| 1    | Super Tuesday  | 2               |
| 2    | Ralph Nader    | 4               |
| 3    | George W. Bush | 2               |
| 4    | caucus         | 3               |
| 5    | John Kerry     | 2               |
| 6    | insurance      | 2               |
| 7    | RIAA           | 2               |
| 8    | Howard Dean    | 3               |
| 9    | shooting       | 2               |
| 10   | IRS            | 2               |
| 11   | Dubya          | 2               |
| 12   | virtue         | 2               |
| 13   | Russia         | 2               |
| 14   | United States  | 2               |
| 15   | Iowa caucus    | 2               |
| 16   | WMD            | 6               |
| 17   | Saddam Hussein | 2               |
| 18   | wisdom         | 2               |
| 19   | apathy         | 2               |
| 20   | crime          | 2               |
| 22   | guns           | 3               |
| 22   | republican     | 3               |
| 23   | democrat       | 2               |
| 24   | crime          | 2               |
| 25   | Iran           | 4               |
| 26   | Clinton        | 2               |
| 27   | democrat       | 2               |
| 28   | campaign       | 4               |
| 29   | gay marriage   | 2               |
| 30   | WMD            | 2               |

**Table 1.** Top 30 buzzwords in the order of their *buzzness*

Caucus”), while ignoring the persistently bursty term “John Kerry.” It may be debatable how to treat such persistently bursty terms. After all, “John Kerry” can be considered as a buzzword of much longer span that associates with a series episodes. Is there any case where the persistently bursty terms should not be considered as buzzwords ? If so, how do we differentiate one case from another ? We leave the problem of dealing with those persistently bursty terms as a future research task.

## 5. Related Work

Our work has many related areas including time series analysis and sequence mining [3, 6], queuing theory for network traffic [8], text mining [12, 13], and event tracking [15, 14].

Kleinberg [7] developed a burst event model using infinite-state automaton and proposed a method to compute the optimal state sequences using Bayes procedure. However, the definition of bursty events as a time interval with high states does not accommodate the more complex characteristics of buzz we try to identify. We developed a new model for detecting buzz using his bursty events as candidates of buzzwords.

The *Yahoo! Buzz Index* counts the percentage of the total number of people searching for a specific query term (or subjects) posted at the Yahoo search engine [1] collected from their search log files. Based on the buzz scores, they

| Interval            | Buzzwords      |
|---------------------|----------------|
| 20040110 – 20040113 | Iran           |
| 20040111 – 20040112 | shooting       |
| 20040111 – 20040112 | apathy         |
| 20040119 – 20040121 | caucus         |
| 20040119 – 20040120 | Iowa caucus    |
| 20040119 – 20040120 | John Kerry     |
| 20040120 – 20040121 | George W. Bush |
| 20040120 – 20040121 | Dubya          |
| 20040120 – 20040121 | United States  |
| 20040121 – 20040122 | RIAA           |
| 20040128 – 20040129 | Russia         |
| 20040128 – 20040129 | Saddam Hussein |
| 20040128 – 20040129 | democrat       |
| 20040128 – 20040129 | Clinton        |
| 20040129 – 20040130 | insurance      |
| 20040201 – 20040206 | WMD            |
| 20040202 – 20040204 | republican     |
| 20040202 – 20040205 | campaign       |
| 20040203 – 20040204 | crime          |
| 20040218 – 20040220 | Howard Dean    |
| 20040218 – 20040219 | virtue         |
| 20040218 – 20040220 | guns           |
| 20040219 – 20040220 | IRS            |
| 20040219 – 20040220 | wisdom         |
| 20040219 – 20040220 | crime          |
| 20040219 – 20040220 | democrat       |
| 20040220 – 20040223 | Ralph Nader    |
| 20040302 – 20040303 | Super Tuesday  |
| 20040311 – 20040312 | gay marriage   |
| 20040325 – 20040326 | WMD            |

**Table 2.** Top 30 buzzwords in the order of their time interval

identify the *leaders*, subjects with highest buzz scores, and the *movers*, subjects with the greatest percentage increase in buzz score from one day to the next. The movers are potentially new buzzwords gaining momentum. The buzz score fundamentally differs from our *buzzness* score (measured by the degree of concentration) in that their buzz terms are essentially most frequent query terms of any given day without taking into account the duration of the buzz.

Recently, blogspace is becoming an active area of research, reflecting the growing popularity in practice [9, 5]. [9] studied evolution of community in blogspace. They view link creation as temporal phenomenon, and developed time graphs to extend the traditional notion of an evolving directed graph over time. They developed algorithms for time-dense community tracking. [5] studies information and topic propagation using blogspace as an example domain. They model the topic propagation at both macroscopic (or corpus) level and microscopic (or individual to individual) levels. Though they deal with topics extracted from the text of blog postings, their main focus is to extract the network of information propagation. They propose an algorithm that induces the underlying propagation network from a sequence of posts.

## 6. Conclusion

In this paper, we presented a formal model of buzz and proposed an algorithm to detect them from a text document stream. Our buzz detection algorithm computes the degree of concentration of occurrences of a term at a given time interval as a measure of *buzzness* of the term. We proposed a formal method of computing the degree of concentration. Our algorithm is experimentally verified on blog postings and the results show that the method is highly promising in detecting buzz.

## References

- [1] Yahoo! buzz index. <http://buzz.yahoo.com>.
- [2] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *Proc. of the International WWW Conference*, 2002.
- [3] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall, 1996.
- [4] D. Gruhl, D. Chavet, D. Gibson, J. Meyer, P. Pattanayak, A. Tomkins, and J. Zien. How to build a webfontain: an architecture for very large-scale text analytics. *IBM Systems Journal*, 43(1):64–77, 2004.
- [5] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proc. of the International WWW Conference*, 2004.
- [6] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, 2001.
- [7] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.
- [8] L. Kleinrock. *Queueing Systems*, volume 1. Wiley, 1975.
- [9] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proc. of the International WWW Conference*, 2003.
- [10] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
- [11] K. Nigam and M. Hurst. Towards a robust metric of opinion. In *Proc. of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2004.
- [12] R. Swan and J. Allan. Extracting significant time-varying features from text. In *Proc. of the International Conference on Information Knowledge Management*, 1999.
- [13] R. Swan and D. Jensen. Timemines: Constructing timelines with statistical models of word usage. In *Proc. of the SIGKDD Workshop on Text Mining*, 2000.
- [14] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving text categorization methods for event tracking. In *Proc. of the SIGIR International Conference on Information Retrieval*, 2000.
- [15] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *Proc. of the SIGIR International Conference on Information Retrieval*, 1998.