

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220907173>

Bursty Feature Representation for Clustering Text Streams

Conference Paper · April 2007

DOI: 10.1137/1.9781611972771.50 · Source: DBLP

CITATIONS

74

READS

125

4 authors, including:



Zhang qi

Dalian University of Technology

21 PUBLICATIONS 1,719 CITATIONS

SEE PROFILE



Kuiyu Chang

LinkSure China Holding Private Limited

70 PUBLICATIONS 1,093 CITATIONS

SEE PROFILE



Ee-Peng Lim

Singapore Management University

441 PUBLICATIONS 8,727 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Recommender systems [View project](#)



WHOWEDA: A Warehousing System for Web Data [View project](#)

All content following this page was uploaded by [Kuiyu Chang](#) on 03 June 2014.

The user has requested enhancement of the downloaded file.

Bursty Feature Representation for Clustering Text Streams

Qi He Kuiyu Chang Ee-Peng Lim Jun Zhang

{qihe0001, ASKYChang, ASEPLim, JZhang}@ntu.edu.sg

School of Computer Engineering, Nanyang Technological University
Block N4, Nanyang Avenue, Singapore 639798

Abstract

Text representation plays a crucial role in classical text mining, where the primary focus was on static text. Nevertheless, well-studied static text representations including TFIDF are not optimized for non-stationary streams of information such as news, discussion board messages, and blogs. We therefore introduce a new temporal representation for text streams based on bursty features. Our bursty text representation differs significantly from traditional schemes in that it 1) dynamically represents documents over time, 2) amplifies a feature in proportional to its burstiness at any point in time, and 3) is topic independent. Our bursty text representation model was evaluated against a classical bag-of-words text representation on the task of clustering TDT3 topical text streams. It was shown to consistently yield more cohesive clusters in terms of cluster purity and cluster/class entropies. This new temporal bursty text representation can be extended to most text mining tasks involving a temporal dimension, such as modeling of online blog pages.

1 Introduction

The performance of a text mining solution often depends critically on the underlying text representation. Modern information sources including news, discussion boards, chat messages, and blogs manifest themselves as text streams/sequences of chronologically ordered documents. With the additional temporal dimension in text streams, classical document representations that have been optimized for static text mining problems (e.g., text clustering and text classification) are poised for a major revamp.

The classical text representation, known as the bag-of-words or Vector Space Model (VSM) [12], represents a document as a vector; with each element denoting the weight/importance associated with a feature in the document. Popular weighting methods include binary, term frequency (TF), and term frequency - inverse document frequency (TFIDF). The feature space typically includes raw words, word stems, phrases, extracted named entities, etc.

The static VSM is not designed to meaningfully model a transition from one semantic context to another over time, not to mention representing evolving trends in text streams. Specifically, a feature in static VSM could refer to completely different topics at various

points in time, e.g., most occurrences of the word feature “war” in news collections *circa* 1998 refer to “the war between NATO and the Serbs”, whereas the same word feature found in news collections *circa* 2002 mostly refers to the “war between US and Iraq” or “war on terrorism”. Grouping words into phrases (e.g., n-grams) as new features and assigning part-of-speech tags to each words can improve the semantic meaning somewhat, but neither approach takes into consideration the time dimension of text streams.

An up and coming topic is usually accompanied by a sharp rise in the reporting frequency of some distinctive features, known as “bursty features”. These bursty features could be used to more accurately portray the semantics of an evolving topic. Figure 1 illustrates the effectiveness of using top bursty features to represent two separate topics. Had we used the usual feature selection and weighting scheme, the word features “Gingrich” and “Newt” frequent in both related but different topics would turn up almost equally important for representing documents of these two topics.

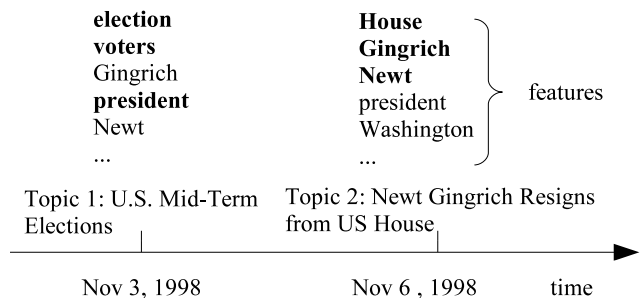


Figure 1: Frequent features of two topics (bursty features shown in bold).

We therefore propose a new text stream representation model, called **bursty feature representation**, which can emulate sophisticated temporal and topical behaviour via bursty features, as illustrated in Figure 1. In our model, a burst corresponds to a phenomenon in which a large amount of text content about a particu-

lar topic is generated over a relatively short time period. Our model varies the bursty features' weights of documents with respect to its publication date. Unlike static document representations, this model dynamically represents a document over time, i.e., a document representation is fully dependent on its publication date. Such a dynamic representation is particularly suitable for modeling text streams, especially for information sources grouped by topics. To the best of our knowledge, we are unaware of any similar prior work.

The remainder of this paper is organized as follows. Section 2 describes related work. In Section 3, we cover the background knowledge of document representation along with the motivation for our bursty feature representation. In Section 4, we describe how to identify bursty features, and subsequently define our self-boosting model for bursty feature representation of text streams. Experimental clustering results are presented in Section 5. Finally, conclusions are drawn and future research directions are identified in Section 6.

2 Related Work

This work is largely inspired by three broad and overlapping research fields.

First, a significant number of recent work has modeled a topic in text streams as a “burst of activities” [7]. Many practical applications have been developed as a result of research in this field. Babcock et al. [1] present a query operator scheduling strategy to minimize runtime memory usage during bursty time periods. Kumar et al. [9] applied Kleinberg’s algorithm [7] to identify bursty communities in a Weblog graph. Mei et al. [11] further summarized the evolutionary thematic patterns of a text stream by identifying bursty behavior. Our approach shares some common ideas as the work of Fung et al. [6], who clustered bursty features and organized them into different bursty events. However, our work is distinct from [6] in two aspects: 1) we use bursty features combined with static features to completely represent each and every document in the text stream, and 2) as a proof-of-concept, we cluster actual documents instead of words, i.e., we find groups of topical bursty documents instead of groups of topical bursty features.

Second, our work is motivated by research in the field of Topic Detection and Tracking (TDT). In TDT, a large amount of research has previously been conducted on identifying emerging topical trends, i.e., detecting new events [2, 8, 13, 16, 17] and tracking topics [3, 5]. However, the vast majority of TDT research does not differentiate trivial topics from bursty topics. Neither do they utilize the time interval information of text streams. For example, the online event detection model proposed by Kumaran et al. [8] simply used time to

obtain the document arrival order. In this paper, we emphasize the importance of time by incorporating it directly into the document representation. Our grouping of similar documents in time and content into corresponding bursty topics can be viewed a new kind of topic tracking and identification model in TDT.

The third inspiration of this work comes from tackling the “curse of dimensionality” problem [4] while processing the hundreds of thousands of unique features in text mining. Documents in very high-dimensional space are almost equally far away from one another, making classical similarity measures like Euclidean distances less discriminative. Thus, an important preprocessing step in most text mining tasks is to reduce the dimensionality or feature space. By selecting only bursty features, as will be shown later, we are in fact reducing the dimensionality of the problem space. Another interesting work by Yu et al. [18] studied the correlation between features with respect to class concepts, with the goal of removing redundant features. This is different from our work which models the dynamic association of word features and documents to topics (classes) utilizing the additional temporal dimension. A detailed survey of text feature selection can be found in [10, 15].

3 Background and Terminologies

Let D be a corpus of text streams with N documents and T be the time period spanned by D . Let F be the complete static VSM features space with $|F| = M$.

3.1 Static VSM As shown in Figure 2, static VSM creates a document vector from a raw text document in two steps: (1) applying text preprocessing such as stopword removal and stemming, and (2) assigning weights to text features as vector elements.

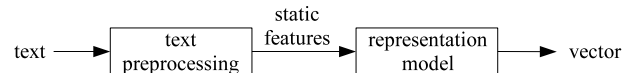


Figure 2: Creating a static VSM model from raw text.

In static VSM, each document \mathbf{d}_i is represented by a vector of M feature weights,

$$\mathbf{d}_i = [d_{i1}, d_{i2}, \dots, d_{iM}]^T,$$

where d_{ij} is the weight of the j -th feature for document \mathbf{d}_i and T denotes the matrix transpose. The weight of a feature determines how much it contributes to the overall document vector \mathbf{d}_i .

A document representation typically involves:

- Feature definition: determine the type of features

to be derived from a document, e.g., term feature type, title term feature type.

- Feature selection/transformation: select and transform features, e.g., both rarely used and overly common term features may be discarded.
- Feature weighting: assign weight to a feature type based on a given formula, e.g., binary, TF, and TFIDF.

3.2 Motivation for Bursty Topic Representation Bursty topics can be emphasized by considering only certain time windows and bursty features. For example, the word feature “hurricane” from topic “Hurricane Mitch” shown in Figure 3 overlaps significantly with the topic document frequency. In this example, a

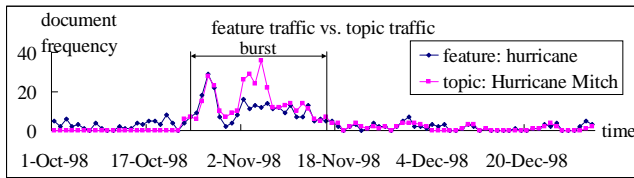


Figure 3: Traffic overlap between a bursty topic and its feature “hurricane”.

total of 401 documents contain the feature “hurricane”, among which 97 are outside of the “Hurricane Mitch” topic. If “hurricane” were the only feature discriminating this topic from the rest, the corresponding precision would be $(401 - 97)/401 = 75.81\%$. From Figure 3, if we only take into account the bursty period of word feature “hurricane” lasting from 24-Oct to 16-Nov, only 10 out of the 260 documents containing the word feature “hurricane” are off-topic, thereby yielding a 20% improvement in precision at $(260 - 10)/260 = 96.15\%$!

The above simple example illustrates the benefit of using a single bursty feature restricted to certain bursty time periods over the whole life span of a topic. If more bursty periods and their associated bursty features are identified, they could collectively improve the overall distinctiveness of each topic.

Such discriminative features with corresponding bursty life spans are called “bursty features” in this paper. To find bursty features, the burstiness of every word feature in the corpus with respect to all topics will have to be computed. In this example, we would need to determine a set of features for the “Hurricane Mitch” topic, and their corresponding bursty life spans. This is a challenging problem because the i -th document in a text stream now has a dynamic vector representation $\mathbf{d}_i(t)$ that depends on time stamp t . In the next

section, we shall present our proposed bursty feature representation to this challenging problem.

4 Bursty Feature Representation

We now describe our bursty feature representation that combines burstiness with static feature weights. Representing a document with bursty features involves two major steps: (1) identifying bursty features, and (2) representing documents using bursty features/weights, as shown in Figure 4.

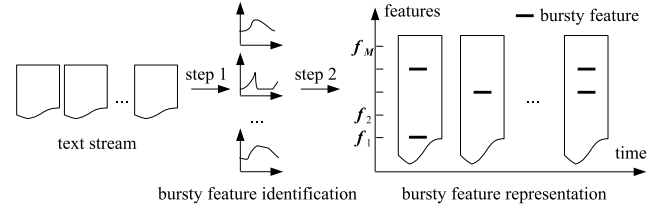


Figure 4: An overview of bursty feature representation.

In Figure 4, a document is assigned bursty weights depending on its time stamp t . The same raw document may have different bursty feature representations at two different time points $t_i \neq t_j$.

In Section 4.1, we will first describe how bursty features can be identified. This is followed by the bursty feature representations in Section 4.2.

4.1 Bursty Feature Identification Bursty feature identification from text streams have recently been investigated by a number of researchers [6, 7, 19]. Since the goal of this paper is to utilize bursty features and not to develop a new bursty feature identification algorithm, we simply adopt Kleinberg’s [7] 2-state finite automaton model to identify bursty features, as shown in Figure 5.

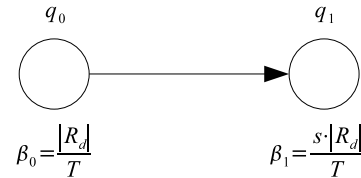


Figure 5: A 2-state finite automaton model for identifying bursty features.

There are two states q_0 and q_1 in the finite automaton model A of Figure 5. For every feature f in a text stream, when A is in state q_0 at time point t , it has a low emission rate $\beta_0 = |R_d|/T$, where $|R_d|$ is the size of all relevant documents containing f over the whole

time range T of the text stream. When A is in state q_1 at time t , the rate is increased to $\beta_1 = s \cdot |R_d|/T$, where $\beta_1 > \beta_0$ because $s > 1$. In other words, we have defined two emission states for each feature f : one with the average document frequency over the whole time duration, and another one with a higher document frequency. The larger the number of documents containing f at time point t , the higher the likelihood of f being identified as a bursty feature at t .

We compute the burstiness of each feature f over all topics and over the full time period of the text stream, i.e., retrospective analysis. Some features may not be bursty at all (zero burst), while others may induce multiple bursty periods. The formal definition of a “bursty feature” is given below.

DEFINITION 4.1. (bursty feature) *If a feature f_i has at least one burst, it is a **bursty feature** with bursty weight w_i and bursty period p_i .*

In Kleinberg’s algorithm, the bursty weight is defined as the cost improvement incurred by assigning state q_1 over time period p_i rather than state q_0 .

Figure 6 plots an example of the bursty feature “impeachment” with two bursts over the entire text stream. The first burst persists from 3-Oct to 10-Oct, 1998 (8 days) with a bursty weight of 13.6775, whereas the second and larger burst spans 16 days from 7-Dec to 22-Dec, 1998 with a bursty weight of 96.0530.

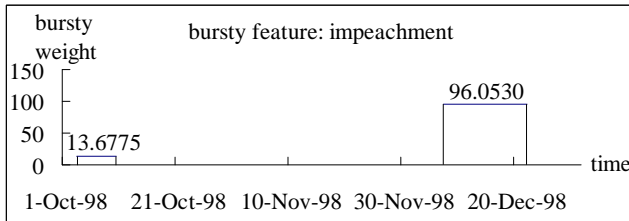


Figure 6: Example of a bursty feature “impeachment”.

4.2 Bursty Feature Representation In practice, due to the sparsity of bursty features and the relatively short period under consideration, a bursty-feature-only representation frequently degenerates into a zero-vector. To understand this phenomenon, just consider any document that either has no bursty features or whose bursty features happen to be all dormant at time t . Clearly, its corresponding bursty feature representation becomes a zero vector, making any similarity comparisons to it meaningless. Therefore, to overcome the zero-vector problem with the pure bursty feature representation, we propose a self-boosting representation that falls back to the static VSM vector in the worst case as follows.

Let B denote the bursty feature space and $B \subseteq F$. Let FP_{ij} denote the static feature weight (i.e., binary weighting) of f_j in document \mathbf{d}_i .

DEFINITION 4.2. (Bursty Feature Representation) *A document $\mathbf{d}_i(t)$ at time t has a bursty feature representation in the form*

$$\mathbf{d}_i(t) = [d_{i1}(t), d_{i2}(t), \dots, d_{iM}(t)]^T,$$

where

$$d_{ij}(t) = \begin{cases} FP_{ij} + \delta w_j, & \text{if } f_j \in B \wedge t \in p_j, \\ FP_{ij}, & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is the burst coefficient.

Here, the role of δ is to combine the sufficiency properties of the static VSM feature space with the discriminative and accuracy properties of bursty features. In other words, bursty features are enhanced or boosted by a factor of δw_j . Non-bursty documents will simply fall back to their static feature representation in the bursty feature space as illustrated in Figure 7. The optimal δ

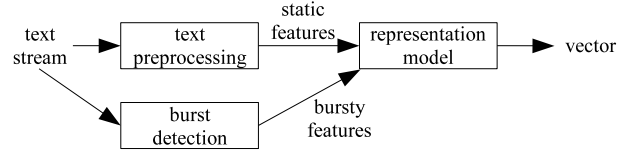


Figure 7: Practical bursty feature representation.

can be determined experimentally via cross-validation.

5 Experiments

5.1 Dataset and Experimental Setup The TDT3 [14] dataset includes 51,183 news articles collected during the three month period of October through December 1998. Among these, 37,526 English articles originated from 8 English sources, and 13,657 Chinese articles came from 3 Chinese sources. We extracted all on-topic English news articles as **TDT3-Eng**, which contains 8,458 articles covering 116 topics.

After stopword removal, 125,468 distinct features remained in **TDT3-Eng**. Among these, 2,160 distinct bursty features with 2,646 bursts were identified as bursty feature space (set B) using the 2-state automaton model described in Section 4.1. We independently selected another 2,160 features (set F) using the document frequency thresholding technique [15].

For a fair comparison, only bursty features in $F \cap B$ are used in our bursty feature representation. Finally we have 1,244 distinct bursty features ($|F \cap B| = 1,244$) with 1,695 bursts, averaging 1.36 bursts per bursty feature.

5.2 Baseline Representation We compare our bursty feature representation to the static binary VSM. Had we used TFIDF, some rare features (with low DF) would be emphasized, and the boosting effect of our bursty feature representation would not be clear.

5.3 Evaluation Metrics Assume that K clusters are generated for dataset D . Let $|k_j|_{C_i}$ denote the number of documents from topic C_i assigned to cluster k_j . Similarly, let $|C_i|_{k_j}$ denote the number of documents from cluster k_j originating from class C_i . We evaluate our clustering results using the known data class labels as follows.

5.3.1 Cluster Purity The purity of cluster k_j is defined by

$$purity(k_j) = \frac{1}{|k_j|} \max_i (|k_j|_{C_i}).$$

The overall purity of a clustering solution is expressed as a weighted sum of individual cluster purities

$$cluster\ purity = \sum_{j=1}^K \frac{|k_j|}{|D|} purity(k_j) = \frac{1}{|D|} \sum_{j=1}^K \max_i |k_j|_{C_i}.$$

In general, the larger the purity value the better the cluster.

5.3.2 Cluster Entropy Cluster entropy measures the diversity of a cluster k_j , and is defined as

$$entropy(k_j) = - \sum_i \frac{|k_j|_{C_i}}{|k_j|} \log \frac{|k_j|_{C_i}}{|k_j|}.$$

The total entropy of a cluster solution is

$$cluster\ entropy = \sum_{j=1}^K \frac{|k_j|}{|D|} entropy(k_j).$$

A good clustering algorithm should have low cluster entropy.

5.3.3 Class Entropy Both cluster purity and entropy measure the homogeneity of a cluster, but neither of them measures the recall of each topic. Thus, we introduce class entropy as follow:

$$entropy(C_i) = - \sum_j \frac{|C_i|_{k_j}}{|C_i|} \log \frac{|C_i|_{k_j}}{|C_i|}.$$

The total class entropy of a cluster solution is

$$class\ entropy = \sum_{i=1}^K \frac{|C_i|}{|D|} entropy(C_i).$$

Ideally, we want the class entropy to be as small as possible.

5.4 Clustering TDT3-Eng We applied K-means ($K = 116$) clustering to **TDT3-Eng**, which comprises 116 topics. Since bursty features are identified based on **TDT3-Eng** itself, the burst coefficient δ is set to 1 to simulate the circumstance in which both static features (in $[0, 1]$) and normalized bursty features (in $[0, 1]$) contribute equally to the representation.

Table 1 lists the 3 evaluation metrics averaged over 10 clustering runs for the binary VSM and bursty feature representations. The metrics are also plotted in Figure 8, which shows the mean, spread (standard deviation) in each direction, and range. From Table

Table 1: Averaged clustering results for **TDT3-Eng** over 10 runs.

<i>representation</i>	<i>cluster purity</i>	<i>cluster entropy</i>	<i>class entropy</i>
binary VSM	0.5750	0.5682	0.8553
bursty feature	0.6149	0.5110	0.7971
Improvement	6.93%	10.06%	6.81%

1, we see that bursty features resulted in clusters with on average 10.06% and 6.81% lower cluster and class entropies, respectively, and 6.93% higher cluster purity. Figure 8 further highlights that bursty feature representation generated more consistent and stable clustering solutions with lower variance and better results in all three metrics.

The results are very encouraging considering that 1) many of the topics in **TDT3-Eng** are small (with just a few documents) and non-bursty, and 2) there is a fair amount of overlap in bursty feature space between the various topics.

6 Conclusions and Future Work

In this paper, we introduced a new bursty feature representation for highlighting the temporally important features in text streams. **The model builds on the classical VSM model and adds an adjustable weight to features considered bursty.** With this bursty feature representation, a document may have very different bursty feature representations at different points in time, i.e., the bursty document vector is dynamic and dependent on time. This captures the importance of bursty words at various periods in time. Experimental results on the TDT3 dataset confirmed and quantified the improvements brought about by our bursty feature representation over a binary VSM model.

As discussed in the paper, many challenges await our future work.

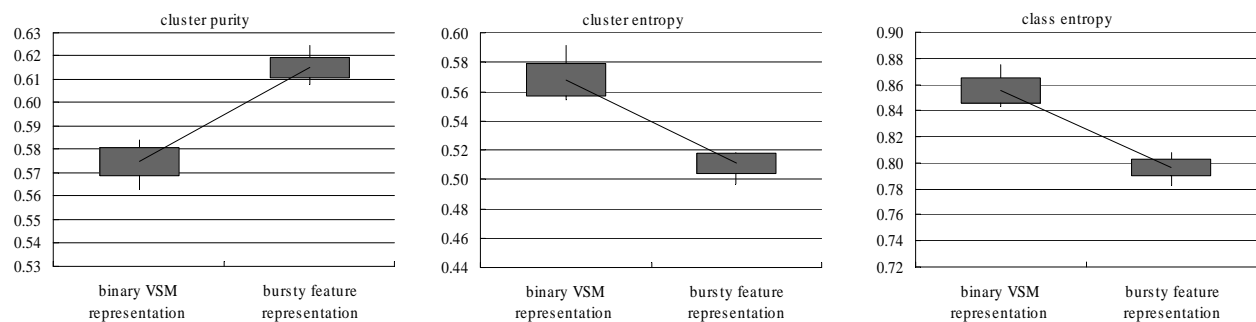


Figure 8: Averaged clustering results for **TDT3-Eng** over 10 runs, showing the mean (end points of line joining the two box plots), spread (box), and range (vertical line).

1. In this work, we have only benchmarked the bursty feature representation against a binary VSM. We would like to compare it with other well-known text representations, especially TFIDF.
2. We have so far only shown the superiority of our bursty feature representation for clustering. We would like to evaluate its suitability for other tasks such as classification and regression.
3. Since we are using Kleinberg's model, we are in fact applying retrospective burst analysis to text streams. In online applications, an alternative online burst analysis model has to be developed in order to analyze and estimate the burstiness of new documents/words as and when they arrive.

References

- [1] B. Babcock, S. Babu, M. Datar and R. Motwani, *Chain: operator scheduling for memory minimization in data stream systems*, In SIGMOD, pp. 253–264, 2003.
- [2] T. Brants, F. Chen and A. Farahat, *A system for New Event Detection*, In SIGIR, pp. 330–337, 2003.
- [3] J. Carthy, *Lexical Chains for Topic Tracking*, PhD thesis, Department of Computer Science, National University of Dublin, 2002.
- [4] K. Clarkson, *An algorithm for approximate closestpoint queries*, In AChf SCG, pp. 160–164, 1994.
- [5] M. Franz, T. Ward, J. S. McCarley and W. J. Zhu, *Un-supervised and supervised clustering for topic tracking*, In SIGIR, pp. 310–317, 2001.
- [6] G. P. C. Fung, Jeffrey X. Yu, Philip S. Yu and H. Lu, *Parameter free bursty events detection in text streams*, In VLDB, pp. 181–192, 2005.
- [7] J. Kleinberg, *Bursty and hierarchical structure in streams*, In SIGKDD, pp. 91–101, 2002.
- [8] G. Kumaran and J. Allan, *Text classification and named entities for new event detection*, In SIGIR, pp. 297–304, 2004.
- [9] R. Kumar, J. Novak, P. Raghavan and A. Tomkins, *On the Bursty Evolution of Blogspace*, In WWW, pp. 159–178, 2005.
- [10] H. Liu and L. Yu, *Toward Integrating Feature Selection Algorithms for Classification and Clustering*, In TKDE, 17(4), pp. 491–502, 2005.
- [11] Q. Mei and C. Zhai, *Discovering evolutionary theme patterns from text: an exploration of temporal text mining*, In SIGKDD, pp. 198–207, 2005.
- [12] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*, Information Processing and Management, 24, pp. 513–523, 1988.
- [13] N. Stokes and J. Carthy, *Combining semantic and syntactic document classifiers to improve first story detection*, In SIGIR, pp. 424–425, 2001.
- [14] <http://projects.ldc.upenn.edu/TDT3/>.
- [15] Y. Yang and J. O. Pedersen, *A comparative study on feature selection in text categorization*, In ICML, pp. 412–420, 1997.
- [16] Y. Yang, T. Pierce and J. Carbonell, *A Study of Retrospective and On-Line Event Detection*, In SIGIR, pp. 28–36, 1998.
- [17] Y. Yang, J. Zhang, J. Carbonell and C. Jin, *Topic-conditioned Novelty Detection*, In SIGKDD, pp. 688–693, 2002.
- [18] L. Yu and H. Liu, *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*, In ICML, pp. 856–863, 2003.
- [19] Y. Zhu and D. Shasha, *Efficient elastic burst detection in data streams*, In SIGKDD, pp. 336–345, 2003.