Bursty Topics Extraction for Web Forums

You Chen
Institute of Computing Technology, Chinese Academy of Sciences
No6, South Road, kexueyuan, Beijing, China
86-10-62600949
chenyou@software.ict.ac.cn

Sen Yang
Institute of Computing Technology, Chinese Academy of Sciences
No6, South Road, kexueyuan, Beijing, China
yangsen@software.ict.ac.cn

XueQi Cheng
Institute of Computing Technology, Chinese Academy of Sciences
No6, South Road, kexueyuan, Beijing, China
cxq@software.ict.ac.cn

ABSTRACT
Many bursty topics which are difficult to summarize and search exist in web forums. Most existing topic detection and tracking (TDT) methods deal with the news stories, but the language used in web forums are much casual, oral and informal compared with news data. In this paper, we present a noise-filtered model to extract bursty topics from web forums using terms and participations of users. Conducting experiments in ShuiMu community we demonstrate the efficiency of our model. Our model not only extracts bursty topics which are better organized for search and visualization, but also discoveries communities corresponding to these topics.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering; H.3.5 [**Online Information Service**]: Web-based service

General Terms: Algorithms, Design, Experimentation

Keywords
Bursty topics, web forums, feature trajectory, frequency segment.

1. INTRODUCTION
Everyday a huge amount of new posts are added to web forums. It is impossible for consumers to read throughout the huge amount of posts. As the availability of such posts increases, the task of summarizing and searching of them becomes increasingly important. Manually monitoring all of them for important topics has become difficulty or practically impossible. In fact, the topic detection and tracking (TDT) [2] community has for many years been trying to come up with a practical solution to help people monitor posts effectively. Unfortunately, the vast majority of TDT solutions proposed for topic detection [3,4,5, 6] are not suitable for posts in web forums, because the language used in web forums are much casual, oral and informal compared with news data used in TDT.

A couple of related work in web forums was reported. Wu et al. [7] used user participation models to extract topics from online discussion. Cheng et al. [8] employed Markov Logic Network on user participation to extract topics. Although user participation which is relatively less noisy can be a set of useful attributes to extract topics, it is not superior to word features. Zhu [9] proposed a model considering not only the word features, but also considering the user activities to extract topics for threaded discussion communities. The model has low performance due to large noisy words existing in web forums.

In this paper, we pay more attention to filtering out noises using burst property of terms. We propose that a term can be characterized and ranked using the concept of frequency segments (sequential occurrences of a term over time). If a term has no burst in its frequency segments, it may be considered as a noise. We use this property to filter out noisy terms. We use participation frequency to find user communities to verify extracted bursty topics. Comparing to terms, participation information is relatively less noisy and easy to process because it only involves in the collection of users' IDs, posting time and posting frequency in web forums.

WIDM'09, November 2, 2009, Hong Kong, China.

2. FEATURE TRAJECTORIES ANALYSIS
2.1 Representation and Weighting
Let T be the duration of posts and F represents the complete word feature space. The representation vector of a term feature is defined as follows:

$$y_f = [y_f(1), y_f(2), ..., y_f(T)]$$

where each element $y_f(t)$ is a measure of feature f at time t, which could be defined using feature frequency

$$y_f(t) = TF_f(t)$$

where $TF_f(t)$ is the frequency of feature f occurring at time t. In order to favor terms appear in titles and entries, we define a term pos-weighted strategy as follows:

$$w_{pos} = \begin{cases} \omega_t & \text{terms in title} \\ \omega_e & \text{terms in entry} \\ \omega_o & \text{otherwise} \end{cases}$$

In web forums, the most informative part of a thread is the title. A thread consists of a title, an entry and some reply posts. A title is the guide of a thread, and an entry is the detail description of a title. The two objects are more informative than other reply posts. It is necessary to emphasize terms existing in titles and entries in web forums. $\omega_t, \omega_e, \omega_o$ are determined empirically. In this paper, we assigned $\omega_t, \omega_e, \omega_o$ as 5, 2, and 1 respectively.

We also characterized users' participation using trajectories, and we define a role-weighting strategy to favor users who post titles. There are two types of users existing in web forums. One is the poster who posts a title or a start post, and the other is the poster who replies to a post. Based on the two types of users, we define a role-weighted strategy as:

$$w_{user} = \begin{cases} \omega_s & author\ of\ start\ post \\ \omega_r & author\ of\ reply\ post \end{cases}$$

ω_s and ω_r are determined empirically. In this paper, we assigned ω_s and ω_r as 5 and 1 respectively.

User trajectories can be defined as:

$$y_u = [y_u(1), y_u(2), ..., y_u(T)]$$

Where $y_u(t)$ is the posting frequency of a user at time t.

2.2 Frequency Segments

There are many methods to divide feature trajectories into segments. It is difficult to determine the size of a time window using this approach. We investigated on many term trajectories and user trajectories, and found that bursts existing in them have one characteristic: the trajectory before or after a burst is discontinuous; and during the bursty period, the trajectory is continuous. Based on the above characteristics, we adapted the following method to divide trajectories into segments. Our method is easy to implement, and it is effectively to extract bursty features which was verified in our experiments.

Given a term feature f, we transform its feature trajectory $y_f = [y_f(1), y_f(2), ..., y_f(T)]$ into the sequence of k segments $X_1, X_2, ..., X_k$. Here a segment is a block of sequence occurrences of a term, and k is the number of blocks. For example, given a feature trajectory (2 1 2 3 0 2 6 5 4 7 3), segments are identified as (2 1 2 3) and (2 6 5 4 7 3). There are two segments in the example, the values of the two segments are $X_1 = 8 = 2+1+2+3$ and $X_2 = 27 = 2+6+5+4+7+3$.

We use two parameters to describe each segment. The two parameters are Sum_{seg} and Dev_{seg}. They are defined as follows:

$$Sum_{seg}[i] = X_i,\ i = 1,2,...,k$$

For each segment, Sum_{seg} is a measure of the absolute strength in the weight of occurrences. The other parameter Dev_{seg} is a measure of the relative strength with respect to the average strength in the weight of occurrences. It is defined as follows:

$$Dev_{seg} = \begin{cases} \min(\Psi, 1.0), & if\ \Psi \geq 0 \\ \max(-1.0, \Psi), & otherwise \end{cases}$$

where $\Psi = \dfrac{Sum_{seg} - mean(Sum_{seg})}{\sigma}$.

$mean(Sum_{seg})$ is calculated as:

$$mean(Sum_{seg}) = \frac{\sum_{i=1}^{k} X_i}{k}$$

σ indicates the standard deviation, it is defined as:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - mean(Sum_{seg}))^2}$$

In the example, there are two segments. The Sum_{seg} of the two segments are 8 and 27, and the Dev_{seg} are -0.72 and 0.72.

As to user trajectories, we use the same transformation with term feature trajectories.

3. MODELING OF TOPICS

3.1 Bursty Features Extraction

For a term feature, we get a vector named sum-dev vector from its frequency segments. The vector is defined as:

$$V_f = [(s_1, d_1), (s_2, d_2), ..., (s_k, d_k)]$$

where $s_i, (1 \leq i \leq k)$ is the value of Sum_{seg} on the i^{th} segment, and $d_i, (1 \leq i \leq k)$ is the value of Dev_{seg} on the i^{th} segment.

We generated a collection of bursty terms using thresholds of $Sum_{seg} > 110$ and $Dev_{seg} > 0.3$. As to each term feature, we scored it with the function:

$$score(f_i) = \alpha * \log(Sum_{seg}) + Dev_{seg}$$

Where α is determined empirically. We assigned α as 0.16 in our experiments, and then we used the scores to rank the bursty terms.

As to core users extraction we use the same strategy with bursty feature detection. We used thresholds of $Sum_{seg} > 20$ and $Dev_{seg} > 0.35$ to select core users in our experiments.

3.2 Measure of Similarity

We measure the similarity between two features from two aspects: feature correlation and post overlap.

For feature correlation, we calculate the similarity of feature distribution. We used the simple function to compute the similarity in deviation patterns of terms. That is given deviation vectors $D_{f_i} = (d_1, d_2, ..., d_k), (i\ is\ the\ order\ of\ term)$ for two term features f_1 and f_2, the similarity between f_1 and f_2 are calculated as $L(f_1, f_2) = \dfrac{< Df_1 - mean(Df_1), Df_2 - mean(Df_2) >}{\| (Df_1 - mean(Df_1)) \| \quad \| (Df_2 - mean(Df_2)) \|}$

where mean(X) denotes the mean value of X, <X, Y>denotes the inner product of X and Y, and ||X|| denotes the norm of X.

Next, we define the overall similarity among a set of features R as follows:

$$L(R) = \min_{\forall f_i, f_j \in R} L(f_i, f_j)$$

Post overlap between two features is also an important factor to measure their similarity. Let M_i be the set of all posts containing feature f_i. Given two features f_i and f_j, the overlapping post set

containing both features is $M_i \cap M_j$. Intuitively, the higher the $|M_i \cap M_j|$, the more likely that f_i and f_j will be highly correlated. We define the degree of post overlap between two features f_i and f_j as follows:

$$p(f_i, f_j) = \frac{|M_i \cap M_j|}{\min(|M_i|, |M_j|)}$$

We define the overall overlap among a set of features R as:

$$p(R) = \min_{\forall f_i, f_j \in R} p(f_i, f_j)$$

As to similarity between users, we use the same function.

3.3 Unsupervised Bursty Topic Detection

Given bursty features $f_i \in F$, the goal is to find high correlated features from F. The set of features similar to f_i can then collectively describe a topic. We need to find a subset $R_i \subset F$ that minimizes the following cost function:

$$C(R_i) = \frac{1}{L(R_i)p(R_i)}$$

We use a threshold δ_L as the minimum value of $L(R)$, and δ_p as minimum value of $p(R_i)$. Upon these two thresholds, we get the maximum value of $C(R_i)$ as $C_{\max}(R) = 1/(\delta_L * \delta_p)$.

Our unsupervised bursty topic detection algorithm is described in Figure 1.

Unsupervised Bursty Topic Detection

Input: Bursty features F, post index for each feature

1: Sort features in descending score order:

$Score_{f1} \geq Score_{f2} \geq \cdots \geq Score_{f|F|}$

2: k=0

3: for $f_i \in F$ do

4: k=k+1

5: Init: $R_i \leftarrow f_i$, and $F = F - f_i$

6: while F not empty do

7: $m = \arg\max_m C(R_i \cup f_m)$

8: if $C(R_i \cup f_m) < C_{\max}(R)$ then

9: $R_i \leftarrow f_m$ and $F = F - f_m$

10: else

11: break while.

12: end if

13: end while

14: Output topics

15: end for

Figure 1. Algorithm of bursty topic detection

4. EXPERIMENTS

The data set used in our experiments is from the "NewsExpress" board on the ShuiMu community [1, 9], which is one of its most popular boards. All posts during Mar.1, 2008 and Mar.10, 2008 are downloaded. The details of the data set are depicted in Table 1. The data set was labeled by Zhu et al. [9]. The raw post data were parsed and post properties were extracted, including: the timestamp, the author, the title and the content. The thread relations of posts are also extracted.

Table 1. Description of the data set

Name	Number
Topics	2,980
Vocabulary	45,961
Threads	7,263
Posts	67,967
users	5,394
Word token	1,651,556

There are totally 67,967 posts of 7,263 threads in the data set, averagely 6796.7 posts and 726.3 threads every day. There are 2,980 topics existing in the data set, and among them 246 topics are bursty. A majority of the topics (more than 2000) consists of only one thread, but the bursty topics which consist of more than one threads cover over 62% of all posts. That is to say, although there are a lot of topics, most of the posts are discussing only a small number of bursty topics. This shows the necessity of bursty topic detection in web forums.

We used two thresholds $Sum_{seg} > 110$ and $Dev_{seg} > 0.3$ to select bursty terms which were depicted in Figure 2.

The number of the bursty terms selected by the thresholds was 2,591. That is to say, we only need to process 2,591 terms instead of 45,961 terms to extract bursty topics. Our method filtered a large number of noisy and unimportant terms. As to the 2,591 terms, we ranked them using the following function:

$$score(f_i) = \alpha * \log(Sum_{seg}) + Dev_{seg}$$

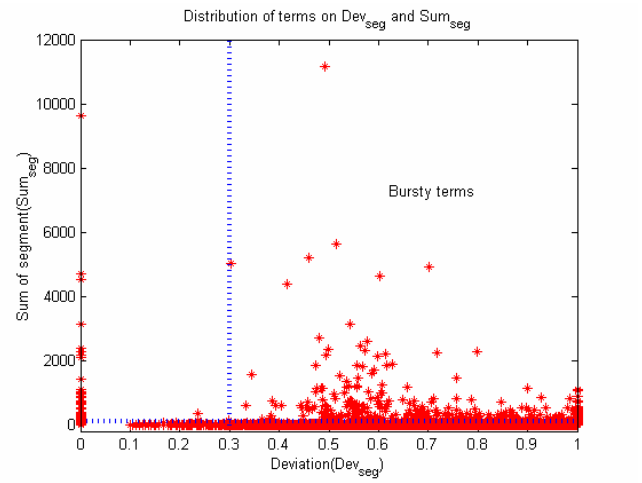

Figure 2. Distribution of terms

We also used two thresholds $Sum_{seg} > 20$ and $Dev_{seg} > 0.35$ to select core users which were depicted in Figure 3.

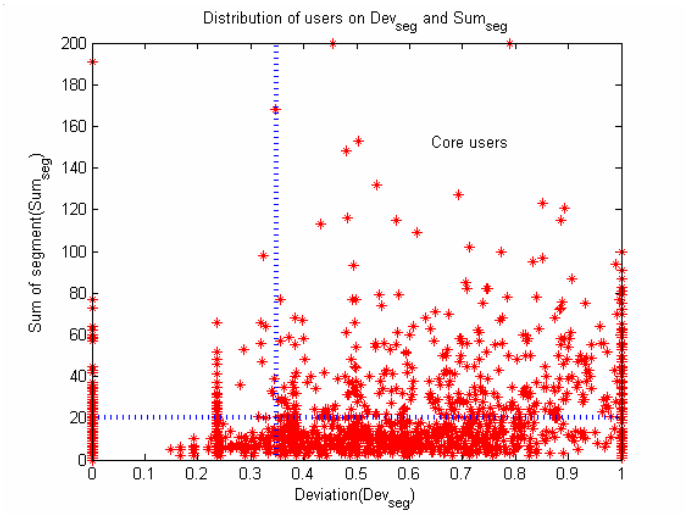

Figure3: Distribution of users

The number of the core users selected by the thresholds was 390. Although there are 5,394 users in data set, the majority of the users are unimportant.

In order to evaluate the performances of our bursty topics detection algorithm, we conducted several experiments on the data set. The experimental results were depicted in Figure 4.

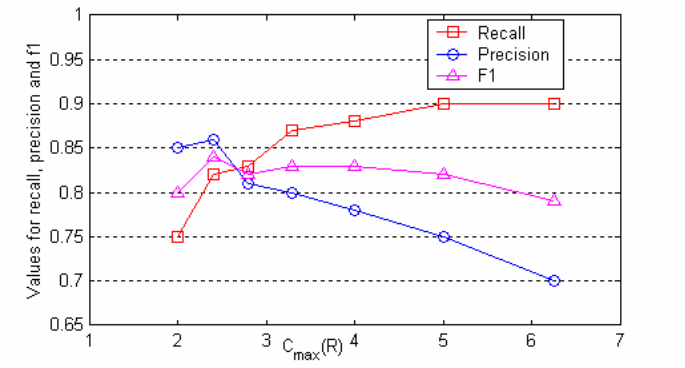

Figure 4. Precision, recall and F1 of bursty topic detection algorithm for different thresholds of $C_{\max}(R)$

For different values of $C_{\max}(R)$, we got recalls, precisions and F1 for our detection algorithm. Through the figure, we can see that when $C_{\max}(R)$ was assigned as 2.4, the algorithm performed best. The best values for recall, precision and F1 are 0.86, 0.82 and 0.84 respectively. The results showed that the majority of bursty topics existing in the data set can be detected by our algorithm. It is also verified that he bursty terms extracted by two parameters Sum_{seg} and Dev_{seg} are important attributes to extract bursty topics.

5. CONCLUSIONS

This paper presents a novel model to extract bursty topics and user communities from web forums. We proposed a concept of frequency segments to transform each trajectory into segments. Based on the frequency segments, we characterized terms or users with two parameters Sum_{seg} and Dev_{seg}. Terms and users can be ranked by using the two parameters. Based on the ranking values, we filtered noisy terms form data collections. Bursty topic detection algorithm was conducted on bursty terms and core users to extract bursty topics. Experiments were conducted on the corpus of web forums. As a result, we could successfully detect a set of bursty topics which were describe by bursty terms.

In future, seeking to predict the future dynamics of topics would be interesting. We plan to solve this problem from core users and their communities. Nevertheless, we believe our simple and effective method will be useful for the initial exploratory analysis of posts streams.

6. ACKNOWLEDGMENTS

This study is supported by the National 863 Program of China under Grant No. 2007AA01Z438; 2006AA01Z452, the National Grand Fundamental Research 973 Program of China under Grant No. 2004CB318109; 2007CB311100.

7. REFERENCES

[1] http://www.newsmth.net/

[2] K. Zhang, J. Li and G. Wu. “New Event Detection Based on Indexing-tree and Named Entity”. In Proc. of ACM SIGIR’07, pages215-222, 2007

[3] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In SIGIR, pages 28–36, 1998.

[4] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In SIGIR, pages 424–425, 2001.

[5] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In SIGKDD, pages 688–693, 2002.

[6] Q. He, K. Chang, and E.-P. Lim. A model for anticipatory event detection. In ER, pages 168–181, 2006.

[7] Zhi-Li Wu, and Chun-hung Li. Topic Detection in Online Discussion using Non-Negative Matrix Factorization. In IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2007

[8] Victor Cheng, and C.H.Li. Topic Detection Via Participation using Markov Logic Network. Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 2007.

[9] Mingliang Zhu, Weiming Hu, and Qu Wu. Topic Detection and Tracking for Threaded Discussion Communities. In IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2008