Detect Events on Noisy Textual Datasets

Sen Yang, Xueqi Cheng, You Chen, Jin Zhang, Hongbo Xu
Institute of Computing Technology, CAS
Beijing, China
{yangsen, chenyou, zhangjin, hbxu}@software.ict.ac.cn, cxq@ict.ac.cn

Gaolin Fang
Tencent Research
Beijing, China
glfang@tencent.com

***Abstract*—Social media, e.g. Weblog and Internet forum, generate rich historical textual datasets which record lots of valuable events. Automatic event detection tries to discover important and interesting events and their related documents. Existing solutions to event detection, however, are mostly proposed for high quality news stories and may not work well when they are applied to noisy social media datasets, where content quality varies drastically from informative to trivial or even spamming. In this paper, an event detection framework, which directly utilizes burst property of events to filter out noise, is proposed. Experimental results on real dataset from Tencent Internet forum, a popular forum in China, demonstrate the effectiveness of the proposed framework.**

Event detection; burst property; noisy textual dataset

I. INTRODUCTION

Large amounts of user generated contents such as texts, pictures, and videos are published on social media everyday. Automatically detecting events hidden in these contents has become a challenging task in recent years. Generally, non-trivial events are events which occur in certain time periods with unusual high degree of people's attention. The property of events that flourishing in certain time period while being inactive in other periods is named burst property in this paper. For example, a lot of people write blogs about Michael Jackson immediately after his sudden death.

Event detection has already been studied for years as a sub-task of Topic Detection and Tracking (TDT) [1, 2, 3, 5, 8]. In TDT, event is defined as "some unique thing that happens at some point in time" [1]. The retrospective event detection (RED) task in TDT was defined to be the task of discovering all of the hidden events in a corpus of stories. Our event detection task is similar to the RED task. However, content quality of datasets in RED is usually relatively high, while datasets in our problem are from social media and noisy which means that they often contain lots of low quality documents, such as advertisements and general statements.

In this paper, we propose a framework by leveraging both content similarity and burst property to detect events on noisy textual datasets. Our approach is based on traditional TDT methods. First, traditional TDT event detection methods are conducted to get event candidates. Then, burst property of each event candidate is studied and final events are generated based on these candidates. We incorporate our proposed framework to representative TDT event detection methods. The evaluation results based on real-life dataset show that our framework can significantly enhance the performance of event detection on noisy datasets.

II. FRAMEWORK FOR EVENT DETECTION

A. Noise Analysis

Usually, noise in textual datasets from social media is mainly introduced by two reasons: low editorial control and aimlessness of the content of documents. Noise caused by the first reason is usually harmful to most problems relating to text processing. In this paper, we focus on noise caused by the second reason. Uninformative (aimless) documents are used to denote documents which do not contain any events in their content. This kind of documents extensively exists in social media and the editorial quality of them could be high.

Uninformative documents could be divided into three types according to content similarity and whether containing bursty words. Bursty words are words which arise unusually during certain periods. For example, in the several days after the sudden death of Michael Jackson, the word "Jackson" arose unusually. The three types are defined as follows.

Noise_Lack_Similarity (NL_S): document which contains bursty words but is dissimilar to any events (means that the content similarity between a document and an event is lower than a predefined threshold).

Noise_Lack_Words (NL_W): document which possesses high content similarity to some event but contains no bursty word in the burst period of this event.

Noise_Lack_both_Similarity_and_Words (NL_SW): document neither possesses any bursty words nor is similar to any events.

B. Noise-filtering Framework

To effectively detect events on noisy datasets, it is essential to filter out NL_S, NL_W, and NL_SW noise. We propose a novel framework which leverages both content similarity and burst property to filter out these kinds of noise.

First, event candidates are generated using any existing event detection methods. In this step, NL_S and partly NL_SW can be filtered out by introducing a minimal content similarity threshold denoted by THRESHOLD_C to each event candidate. A document will be included in an event

978-0-7695-4012-2/10 $26.00 © 2010 IEEE
DOI 10.1109/APWeb.2010.22

candidate only if the content similarity between the document and the candidate is larger than THRESHOLD_C.

Second, burst property of each event candidate is studied and bursty words are collected. Candidates which have no bursty word are directly discarded and all documents in these candidates are labelled as NL_SW.

Finally, based on detected bursty words, final events are generated within each event candidate. NL_W noise would be filtered out in this step.

C. Description of Burst Property

Generally, a non-trivial event corresponds to an appearance of unusually arising of the amount of related documents. Unfortunately, we could not directly use the burst of document to decide whether an event candidate possesses burst property because we have no idea whether a document does belong to an event before we have gotten the true event. Since the content similarity has been introduced to the inside of each candidate after the conducting of traditional event detection methods, the same word in a candidate could be thought as possessing the same discussion topic. Therefore, it is reasonable to approximate burst of document by burst of words in the study of the burst property of event candidates.

In our proposed framework, automaton model is used to detect bursty words in each event candidate [4]. In [4], automaton model represents a document stream using a finite-state automaton, in which different states correspond to different rates of document arrivals. The onset of state transition between states indicates a birth or death of a burst. The 2-state automaton model [4] is adopted in our approach.

D. Generating Events from Event Candidates

Events are finally generated from event candidates. If an event candidate does not contain any bursty words, all documents in this candidate are labeled as NL_SW. We propose a novel unsupervised event generation algorithm which generates events using bursty words within each event candidate in Fig. 1. MIN_CONF is a user-defined threshold and is used to control the granularity of resulting events.

For two bursty word sets, such as W_i and W_j, we design a novel correlation function named time_confidence based on confidence to measure the burst correlation between them. Time_confidence is defined as

$$T\_Conf\left(W_i, W_j, I_{W_i}, I_{W_j}\right) = \frac{count_{I_{W_i} \cap I_{W_j}}\left(W_i \cap W_j\right)}{count_{I_{W_i}}\left(W_i\right)} . \quad (1)$$

I_w represents burst period of word set W. I_W represents the common burst period of all words in word set W.

Input: all bursty words of an event candidate: $W = (w_1, \dots, w_n)$.

Output: all events from this candidate: set $E = (e_1, \dots, e_m)$.

```
Sort words in W according to their burst scores in descending order:
Tag each word in W as unused.
while there are words in W are tagged as unused do
  is_first_word = true;
  for each word w in W
    if is_first_word = = true then
      Create new event e; Add w to the bursty word set of e; Tag w as used; Set the time interval of e to time intervel of w; Add documents containing w to e; is_first = false;
    else if T_Conf(e, w, I_{w∩e}, I_{w∩e}) > MIN_CONF and
            T_Conf(w, e, I_{w∩e}, I_{w∩e}) > MIN_CONF then
      Add w to the bursty word set of e; Tag w as used; Set the time interval of e to the union of time interval of e and time interval of w; Add documents containing w to e;
    end if
  end for
  Add event e to event set E;
end while
```

Figure 1. Event generation algorithm.

III. EXPERIMENTS

A. Data Collection and Evaluation Methods

We collect 12,177 original messages (have more than 5 replies) ranging from June 1st 2007 to June 30th 2007 from Tencent Internet forum [9] as our experimental dataset.

Our framework is evaluated using standard TDT evaluation method [2]. According to [2], a method may extract any number of bursty events, but is only evaluated on the selected reference events. In our collected dataset, we manually labelled five non-trivial events and the evaluations are conducted on these five reference events.

Three evaluation measures are defined, including traditional precision (p), recall (r), and the F_1 measure (F_1), as in [1]. To measure global performance, overall values for the precision, recall and F_1 measure are calculated by taking weighted average of all values for the corresponding measures.

B. Representation and Parameter Setting Policies

The public natural language processing (NLP) tool [10] is used to parse title and content of each document and collect words. Common Chinese stop words are removed. VSM model is used to represent each document (combining title and content with equal weight) where weight of each word (or named term) is calculated using traditional TF∗IDF scheme [6]. Standard cosine similarity is used to measure the similarity between two term vectors (either a document term vector or a cluster term vector).

THRESHOLD_C, which is used to exclude irrelevant documents in candidate events, is set to 0.1. The jump probabilities of the 2-state automaton model both are set to 0.1. MIN_CONF in events generation algorithm is set to 0.6.

C. Performance Evaluations

Two representative algorithms, namely, K-Means clustering algorithm and the time window event detection

algorithm (WND) [1] are used to demonstrate the effectiveness of our framework.

1) Comparing with K-Means: Clusters are represented as term vectors, which are defined as the arithmetic average of term vectors of all documents within them. The clustering results are considered as detected events. Performances of K-Means and our framework based on K-Means (denoted by F-K-Means) according to different initial cluster numbers both are shown in Fig. 2. We find that, for all initial cluster numbers, performances on F_1 measure of our framework are much better than that of K-Means.

2) Comparing with WND: WND method is one of the best methods in TDT [7]. In WND, the choices of window size and clustering threshold are tricky [1]. Performances of WND and our framework based on WND (denoted by F-WND) on different window sizes and clustering thresholds are shown in Fig. 3.

3) Overall Analysis: The best performances of K-Means, WND, F-K-Means, and F-WND are given in Table I. Based on Fig. 2, Fig. 3, and Table I, we could find that the best performance of WND is much better than that of K-Means. However, the difference between best performances of F-K-Means and F-WND is much less significant. This is mainly because of the directly introducing of burst property to noise filtering in our framework. This explanation also applies to the explanation of the phenomenon that the performances of
F-WND and F-K-Means on different parameter settings are relatively smoother than that of WND and K-Means respectively.

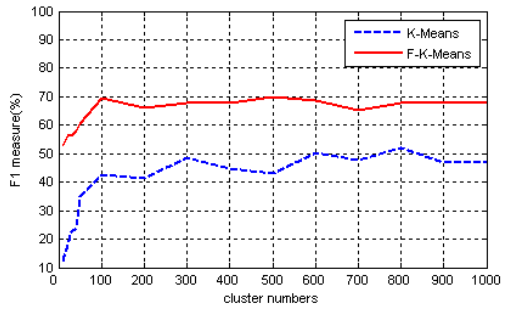

Figure 2. Performances of K-Means and F-K-Means on different initial cluster numbers.

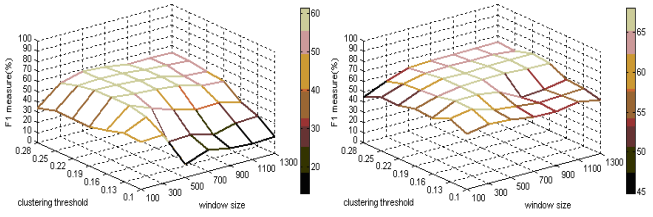

Figure 3. Performances of WND and F-WND on different window sizes and clustering thresholds (The left figure is based on WND and the right one is based on F-WND.)

TABLE I. BEST PERFORMANCES OF K-MEANS, WND, F-K-MEANS, AND F-WND

System	p (%)	r (%)	F_1 (%)
K-Means	44.7480	68.7831	52.1253
WND	66.7590	58.2011	61.4061
F-K-Means	73.8525	66.6667	69.7202
F-WND	71.4716	66.1376	68.0827

IV. CONCLUSIONS

We have proposed an effective framework for event detection on noisy textual datasets. Firstly, we carefully analyzed the potential noise on datasets from social media. Then, a framework which detects events by leveraging both content similarity and burst property was introduced. Extensive experiments were conducted to evaluate the proposed framework and the results show that our proposed framework outperforms baselines on noisy datasets. In the future, we want to pay more attention to utilizing various features on specific social media domains, such as reply and user information on Internet forum, to improve the performance of event detection.

ACKNOWLEDGMENT

This study is supported by the State Key Program of National Natural Science of China (Grant No. 60933005), the State Program of National Natural Science of China (Grant No. 60903139), the 863 Program of China 2007AA01Z438, and the Founds of Tencent Research.

REFERENCES

[1] Y. Yang, T. Pierce, and J. Carbonell. "A study on retrospective and on-line event detection". In Proc. SIGIR'98, pp. 28-36, 1998.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. "Topic detection and tracking pilot study: final report". Proc. of the DARPA Broadcast news Trans. and Understanding Workshop, 1998.

[3] Qi He, Kuiyu Chang, and Ee-Peng Lim. "Analyzing feature Trajectories for event detection". In Proc. SIGIR'07, pp. 207-214, 2007.

[4] Jon Kleinberg. "Bursty and hierarchical structure in streams". In Proc. SIGKDD'02, pp. 91-101, 2002.

[5] Mingliang Zhu, WeiMing Hu, Ou Wu, "Topic detection and tracking for threaded discussion communities", In IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 77-83, 2008.

[6] Gerard Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.

[7] J. Zhang, Z. Ghahramani, and Y. Yang. "A probabilistic model for online document clustering with applications to novelty detection". In Proc. NIPS'05, 2005.

[8] Zhiwei Li, Bin Wang, Mingjing Li, Wei-Ying Ma. "A probabilistic model for retrospective news event detection". In Proc. SIGIR'05, pp. 106-113, 2005.

[9] http://bbs.qq.com/.

[10] http://www.nlp.org.cn/.