Events Detection and Temporal Analysis in Social Media

Yawei Jia(✉), Jing Xu, Zhonghu Xu, and Kai Xing

University of Science and Technology of China, No. 443, Huangshan Road, Shushan District, Hefei 230027, Anhui, China
{ywjia,jxu125,xzhh}@mail.ustc.edu.cn, kxing@ustc.edu.cn

Abstract. In the past few years, event detection has drawn a lot of attention. We proposed an efficient method to detect event in this paper. An event is defined as a set of descriptive, collocated keywords in this paper. Intuitively, documents that describe the same event will contain similar sets of keywords. Individual events will form clusters in the graph of keywords for a document collection. We built a network of keywords based on their co-occurrence in documents. We proposed an efficient method which create a keywords weight directed graph named KeyGraph and use community detection method to discover events. Clump of keywords describing an event can be used to analyse the trend of the event. The accuracy of detecting events is over eighty percents with our method.

Keywords: Event detection · KeyGraph · Co-occurrence · Temporal analysis

1 Introduction

With fast development of social media, such as micro-blog, which becomes the most popular platfom to communicate and express their views. A large amount of data is produced each day, which contains large amount of valuable information. In fact the communication and interactions in social media reflect events and dynamics in real world. We propose a method to mine social media to discover events happened in reality and an algorithm to identify hot events in this paper.

Generally, an event can be described by a set of descriptive, collocated keywords or terms. The mission of event detection is to cluster these topological meaningful keywords into groups. There are several ways to extract and cluster keywords from documents. We might take the document-pivot clustering methods which firstly cluster documents into several groups and then select keywords from the clusters of documents based on some feature selection approaches. However, the association relationship of keywords and the influence of one keyword on another are missed in these methods. In fact, the co-occurrence of terms is very important in event detection. For example, it is meaningless if the terms *Trump*, *Hillary* and *President* appear in three distinguish documents. If they co-occur in documents and we know the conditional probability one term occur

C.-Y. Lin et al. (Eds.): NLPCC-ICCPOL 2016, LNAI 10102, pp. 401–412, 2016.
DOI: 10.1007/978-3-319-50496-4_33

on another, we know more from the constellations of keywords. We build, therefore, a weighted directed graph named KeyGraph to capture the topological information existing among keywords.

In consideration of the importance of source of documents, we innovatively focused the authority of author of documents when keywords are extracted from documents. We try to create a graph of keywords, nodes of which are the keywords, and there exists an edge between keywords if they co-occur in a document. The weight of edge is computed by a probabilistic feedback mechanism. We adopt community detection algorithm adapted from social network analysis algorithm on the graph to discover events. Constellations of terms describing events may be used to track the trend of events.

2 Related Work

The target of event detection is to find a minimal set of keywords that can indicate an event. Kumaran *et al.* showed how performance on new event detection can be improved with using text classification technique [4], and Yang *et al.* adopted several supervised text categorization methods specifically some variants of K Nearest Neighbour algorithm to track events [10]. All of the methods mentioned above are based on document-pivot clustering. In general, all documents are clustered into several groups at first. Then, They select features or terms from the clusters of documents with some feature selection approaches to represent an event. It is worth noting that in the document-pivot clustering approach, keywords as a whole need to be considered to measure the similarity between two documents. Fung *et al.* reported that the most similar documents often belong to different categories, therefore this approach can be biased to the noisy keyword [3].

Li *et al.* [5] proposed a probabilistic model for news event detection, they use a mixture of unigram models to model contents and Gaussian Mixture Model (GMM) to model timestamps, and the parameters are estimated by Expectation Maximization (EM) algorithm. Those algorithms require the number of events. [9] propose a novel sketch-based topic model together with a set of techniques to achieve real-time detection and [11] proposed a novel solution to detect both stable and temporal topics simultaneously from social media data.

3 Keywords Extraction

Let us denote $D = \{d_1, d_2, \ldots, d_n\}$ be the collections of documents and $U = \{u_1, u_2, \ldots, u_i\}$ be the users set of these documents (user refers to the author of documents in this paper). And $W = \{w_{11}, w_{12}, \ldots, w_{ij}, \ldots\}$ is the words set. w_{ij} means the jth word in the ith document. Each word w_{ij} is from a document d_i in documents collection D. This section focus on how to extract keywords from words set. Considering the importance of source of documents, we innovatively take user's authority into consideration. Specifically, we estimate users' authority with an algorithm adapted from the classical PageRank Algorithm, and compute

the keywords tf-idf value. With users' authority and keywords, we can compute a score of candidate keywords. Then, the keywords could be selected from the words set W according to the scores.

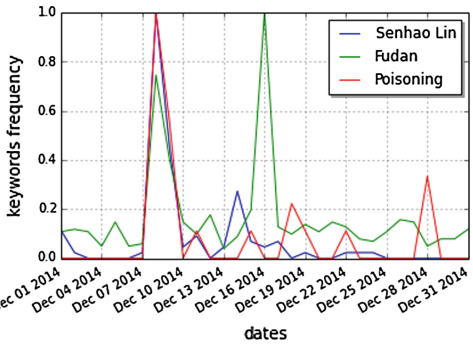

Fig. 1. An example for frequency of keywords associating with hot event.

3.1 User Authority Estimation

Considering that our experiment is conducted on the social network documents dataset, we use the user of social network to introduce user's authority estimation. Social network such as tweeter allow all the registered people to post and share short messages. Since every user has different influence on the public, the contents whose generator has higher authority are easier to be disseminated among social community, and the contents are more likely to be a hot event.

In social network community, if user u_i is interested in the contents which user u_j posts or shares, u_i may follow user u_j and u_i is called a follower of u_j. And u_j does not have to reciprocate by following user u_i. We can model the relationship of users in the social community using a directed graph $G = < U, E >$ where U is the set of users and E is the set of edges between users. There is a directed edge from user u_i to u_j, if user u_i is a follower of u_j. As the directed graph is similar to the web page network in topology, the authority of users can be estimated by the following formula adapted from PageRank algorithm:

$$auth(u_i) = (1 - \alpha) + \alpha \cdot \sum_{u_j \in follower(u_i)} \frac{auth(u_j)}{following(u_j)} \tag{1}$$

In the formula (1), α is a dumping parameter which is introduced by the author in [6]. Its value is usually set to 0.85, which represents the probability that a random surfer of the graph G moves from a user to another. $follower(u_i)$ is the set of users who follow the user u_i. Then we can compute each user's authority

with an iterate algorithm based on the Page-Rank Algorithm [6] with an initial value:

$$auth(u_i) = \frac{1}{|following(u_i)|}$$

3.2 Words Score

The first challenge to detecting event is extracting keywords. During the period within which an emerging event become popular, the frequency of keywords indicating the event will show an upward trend along the time axis. For example, we show the frequency of the keywords describing the event of "*Fudan University Poisoning case*" in Fig. 1, which had taken great attention in China in 2014. It is obvious that the three key words "*Fudan*", "*Senhao Lin*" and "*Poisoning*" happen to coincide to burst during December 7th day to December 11th day in 2014. We use the TF-IDF [7] to define the relative importance of keyword. The tf value of the jth keyword of the ith micro-blog document is computed by:

$$tf_{i,j} = 0.5 + 0.5 \cdot \frac{tf_{i,j}}{tf_{i,j}^{max}} \tag{2}$$

then the idf value of the jth keyword of the ith document is shown as follows:

$$idf_{i,j} = log(\frac{|D|}{1 + |i \in D : j \in i|}) \tag{3}$$

where $|D|$ is the total number of documents. Given tf and idf, the tf-idf value is given by:

$$tfidf_{i,j} = tf_{i,j} \cdot idf_{i,j} \tag{4}$$

With tf-idf value of keyword and users' authority, the score of words w_{ij} is computed by the following equation:

$$score_{i,j} = \sum_{d_i \in D} tfidf_{i,j} \cdot auth(user(d_i)) \tag{5}$$

where $user(d_i)$ here is the author of document d_i.

3.3 Keywords Selection

With the score list of all words, we can select words with higher score as keywords. Intuitively the words describing a hot event will have a high score because the hot event usually could catch the user's attention who has a high authority and have a high tf-idf value for the wide spread. We use the following method based on [1] to compute the cut-off point to identify keywords:

1. First rank the words in descending order of score computed.
2. Compute the *maximum drop* in match and identifies the corresponding drop point.

3. Compute the *average drop* (between consecutive keywords) for all those keywords that are ranked before the identified maximum drop point.
4. The first drop which is higher than the average drop is called the *critical drop*. We returned keywords ranked better than the point of critical drop as candidate keywords.

4 Events Detection

We adopt a community detection algorithm on a keywords graph named KeyGraph to discover events. We build a KeyGraph whose nodes are the keywords and edges are formed between nodes when keywords co-occurs in a document. Generally, keywords co-occur when there is some meaningful topological relationship between them. We can regard the KeyGraph as a social network of relationship between keywords. As is shown in Fig. 2, it is clear that community of keywords are densely linked and there are few links between keywords from different communities.

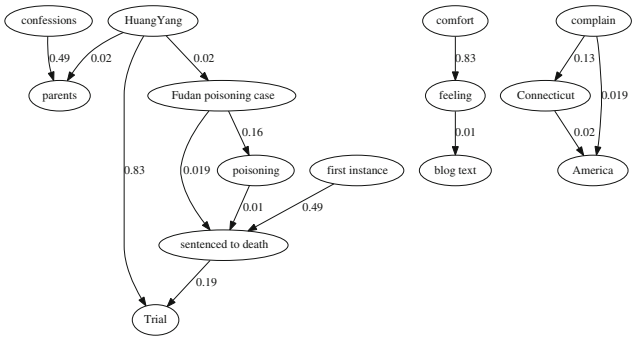

Fig. 2. An example for KeyGraph.

4.1 Building KeyGraph

We build KeyGraph through a multigraph of keywords. Nodes are the keywords and there are n edges between the nodes if keywords co-occur n times in documents. As in Fig. 3, if there is some meaningful topological relationship between keywords, there are many edges between them. We can take advantage of this property to remove some noise in data. Specifically, we repeat the following two steps on each node and edge of the multigraph until nothing can be done.

(a) The number of edges between the two keywords must be larger than some minimum threshold. Otherwise, all of the edges between the two keywords are removed.

(b) The degree of each node in the multi-graph must be equal or larger than the threshold that is set in the rule (a). Or the node will be eliminated from multi-graph.

In short, edges are removed if the keywords associated with nodes co-occur below a minimum threshold and the resulted isolated keywords are removed.

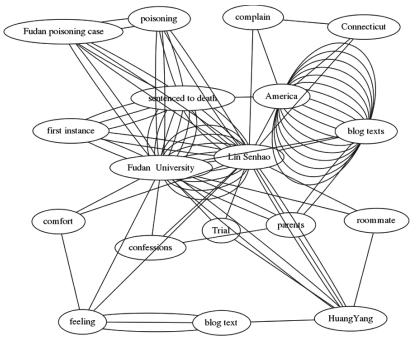

Fig. 3. Example of multi-graph of keywords.

We could build KeyGraph conveniently based on the multi-graph. All nodes in multi-graph are kept in KeyGraph and there is an weighted directed edge from node k_i to k_j if there are edges between nodes k_i and k_j in multi-graph. Here we assume that the weight $c_{i,j}$ is greater than $c_{j,i}$ without generality. The weight $c_{i,j}$ between nodes k_i and k_j can be calculated as shown:

$$c_{i,j} = \log \frac{n_{i,j}/(d_i - n_{i,j})}{(d_j - n_{i,j})/(N - d_j - d_i + n_{i,j})} \cdot |\frac{n_{i,j}}{d_i} - \frac{d_j - n_{i,j}}{N - d_i}| \tag{6}$$

where:

- $n_{i,j}$ is the number of edges between the nodes k_i and k_j in the multi-graph.
- d_i is the degree of node k_i in the multi-graph.
- d_j is the degree of node k_j in the multi-graph.
- N is the total number of nodes.

It is noticed that the first term in the formula will increase as the times of co-occurrences between keywords i and j increase and the second term will decrease as the number of occurrences of a single keyword reduce. Actually, the $c_{i,j}$ is similar to conditional probability $p(k_i|k_j)$ of seeing keywords k_i in a document if k_j exists in the document which reflects the influence one keyword on another. Figure 2 shows an example for KeyGraph.

4.2 Community Detection

We apply community detection techniques adapted from network analysis method to discover events from the KeyGraph. Because the KeyGraph is a

weighted directed graph, we adopt the method proposed in [2]. We first find all fixed size k of clique, for example k-clique (k = 3). Only when the intensity of clique is larger than a threshold value, will the clique be included. Two cliques are defined adjacent if they share k−1 nodes. A community is the union of cliques, in which we can reach any k-clique from any other clique through a series of k-clique adjacencies. Finally the communities of describing, collocated keywords are the discovered events we want.

5 Temporal Analysis

Event always has a temporal characteristics. The events detected by the algorithm should have a trend along the time axis. Basically, hot event would be spread widely and many documents will report the event. Considering the fact that the collocated keywords describing the event would cumulatively increase, we define a binary-valued function:

$$f(k|d) = \begin{cases} 1\,, k \in d \\ 0\,, k \notin d \end{cases} \tag{7}$$

where k is a keyword and d is a piece of document. For the detected event e_i, its trend in time internal $[t_0, t_0 + t]$ is shown as follows:

$$tr^{(t)}(e_i) = \sum_{k \in e_i} \sum_{d \in D^{(t)}} f(k|d) \tag{8}$$

where e_i is the ith event discovered by the algorithm, $D^{(t)}$ is the collection documents in time internal $[t_0, t_0 + t]$ and t_0 is a point time.

For each event e, we could compute its $tr^{(t_j)}(e), j = 1, \ldots, n$ in n series time unit. In order to detect the burst point of $tr(e)$, we compute the cumsum of the series $tr(e)$ as follows:

First, we compute the mean value of $tr^{(t_j)}(e), j = 1, \ldots n$:

$$\overline{X} = \frac{\sum_{i=1}^{n} tr^{(t_i)}(e)}{n} \tag{9}$$

Then, the cumsum is denoted as S_j:

$$\begin{aligned} S_1 &= tr_{(t_1)} \\ S_j &= S_{j-1} + tr_{(t_j)}(e) - \overline{X} \end{aligned} \tag{10}$$

In general, $tr(e)$ added to S_j is positive and the S_j will steadily increase. And if the event occurs at a certain time, the sum value will rapidly increase. A segment of the cumsum chart with an upward slop appears before the burst point, which indicates a period of time where the values tend to be larger than the average. A change in direction of cumsum chart shows that the event bursts after the change point in the cumsum chart. We introduce an algorithm to detect the change point. The estimator of magnitude of the change is defined as follows:

$$S_{diff} = S_{max} - S_{min} \tag{11}$$

Where $S_{max} = \max_{j=1,\ldots,n} S_j$ and $S_{min} = \min_{j=1,\ldots,n} S_j$.

For an event e and its $tr^{(t_j)}(e)$ values in n time units, we perform a bootstrap analysis [8] as follows:

1. Generate a bootstrap sample by randomly reordering the $tr^{(t_j)}(e)$.
2. Based on the bootstrap sample, compute the bootstrap cumsum as shown in the formula (10) denoted as $S_1^{(b)}, \ldots, S_n^{(b)}$.
3. Compute the maximum, minimum and the difference of bootstrap cumsum which are denoted as $S_{min}^{(b)}, S_{max}^{(b)}$ and $S_{diff}^{(b)}$.
4. Compare the original S_{diff} to the bootstrap $S_{diff}^{(b)}$. If S_{diff} is larger than $S_{diff}^{(b)}$, the event e is labelled as a hot event.

The idea behind the bootstrap analysis is that we can estimate how much $S_{diff}^{(b)}$ would vary if no change took place by performing a large number of bootstrap sample. Then We compare the bootstrap $S_{diff}^{(b)}$ value with the S_{diff} of original data so as to assure whether there are change point in the original data.

6 Experiment Analysis

In order to evaluate the performance of the method proposed in this paper, we conduct the experiment on sinaweibo micro-blog documents that we have collected during the twelve month from January to December in 2014. In this section we give a description of dataset on which experiment conducted and then provide the experiment result with analysis.

6.1 Dataset

We crawled the micro-blog documents from the internet. The total dataset has over 70 millions records and each record consists of micro-blog document texts, the generator of a piece of micro-blog document and the timestamp when the micro-blog document was created. Considering the volume of datasets and the nature of events distribution, we partitioned the datasets into twelve timeslots from Jan 2014 to Dec 2014. Each timeslot contains the micro-blog document data posed in one month.

6.2 Experiment Result and Analysis

Compared with English, Chinese must be segmented into words first. We choose to use NLPIR[1] to segment micro-document texts into words. After removing stopwords and non-characters such as emotion symbols, we applied the proposed

[1] A Chinese word segmentation system. http://ictclas.nlpir.org/.

Table 1. The events detected during January through December in 2014.

Date	Keywords	Events description
Jan 2014	{Open,Champion,Final,Won, Women's Single,Dominika Cibulkova,Eugenie Bouchard,Li Na,Australian Open,Azarenka } { Indonesia,volcano,burst}	{Li Na won Australian Open Women's Singles } {Indoesia volcano burst }
Feb 2014	{Portugal,legend super-star,Eusebio,died,Panthers} {President,Ukraine,Viktor Yanukovich,lift, parliamentary,duties }	{The death of Eusebio } {Ukrainian Parliament Deprives Yanukovych Of Presidential }
Mar 2014	{hospital,blood cen-ter,Kunming,reinforcement} {taking the virus,Kindergarten, children,investigation}	{Need to reinforce the blood center of Kunming hospital} {Children of a Kindergarten in Jilin were taken the "spiritual virus"}
Apr 2014	{Star Wars,death,funeral, boy,British,wishes} {wreck,South Korea,staff,escape}	{4-Year-Old Receives Star Wars-Themed Funeral As His Final.} {South Korean ferry disaster}
May 2014	{Gutman,won,the European cham-pionship,UEFA Europa League Cup }	{Gutman won championship in UEFA Europa League Cup}
Jun 2014	{Nanjing,Yangzi,refinery, explosion,fire,apparatus} {Abe,Tokyo,protest,lifted, collective,self-defense right,self-immolation}	{Petrochemical explosion in Nanjing China} {A Japanese committed self-immolation to protest Japan's push to expand defense role}
Jul 2014	{Temporary shelters,period ex-pired,license,Snowden,asylum}	{The period of temporary asylum of Snowden expire.}
Aug 2014	{Earthquake,disaster, emergency,rescue, soldiers,marching,army} {Accident,Investigation, intervention,Kunshan, explosion}	{Troops rush to quake to rescue refugees} {Investigate the accident of Jiangsu Kunshan plant explosion}
Sep 2014	{Entrance Examination,reform,cancel, division,arts,science.} {rural,homestead, occupied,Zhangquan Liu,Chuanming Zhou,rogue}	{the Entrance Examination reform cancel the division of arts and science.} {rural homestead was occupied by rouge Zhangquan Liu and Chuanming Zhou}
Oct 2014	{ Artist, misdeeds, drug, ban, prostitu-tion }	{SARFT of China claims to ban misdeeds artists }
Nov 2014	{The Navy, Colonel, trickster, posing, cheat}	{Laid-off workers posing Navy Colonel cheat two women.}
Dec 2014	{Fudan University, Senhao Lin, poison-ing, sentenced to death}	{Senhao Lin, student in Fudan Univer-sity, was sentenced to death for poisoning his roommate.}

method and algorithm to the dataset. Experiment result showed that is quite efficient with our algorithm as listed in Table 1.

In Table 1, the second column are the collocated keywords that belong to one community and the third column is the description of corresponding event. For each detected event we checked the mainstream media so as to determine whether it really happened in the real world. The accuracy was computed as follows:

$$Accuracy = \frac{\#true\_events}{\#true\_events + \#false\_events} \tag{12}$$

where

– #true_events is the number of events that really happened in real world.
– #false_events is the number of mistaken events by our algorithm.

The experiment result showed that the accuracy is around 80% as is seen in Fig. 4.

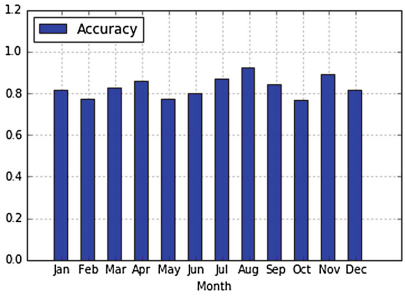

Fig. 4. Accuracy.

For identifying hot events, we compute the cumsum of $tr(e)$ to detect the burst of events. If an event won't become hot, its $tr(e)$ would not burst suddenly. What's reflected in the cumsum chart is that the cumsum chart will be a smooth line. In other words, there won't be change points in cumsum chart. With that we design a bootstrap sample analysis based algorithm to detect the hot events. In the algorithm, for an event, we determine whether it is a hot event by detecting the change point in the cumsum chart. Like the events in the left side of Fig. 5 the cumsum line increase sharply where there is a change point. The change points are detected by our algorithm, the events *Australian Open Women's Champion(Na Li from China won the Champion)* and *Mo Zhang detained for taking drugs* are identified as hot events. On the contrary, the events in the right side of Fig. 5 would not be identitied as hot events because no change points are detected. The experiment results demonstrate that the algorithm is very efficient.

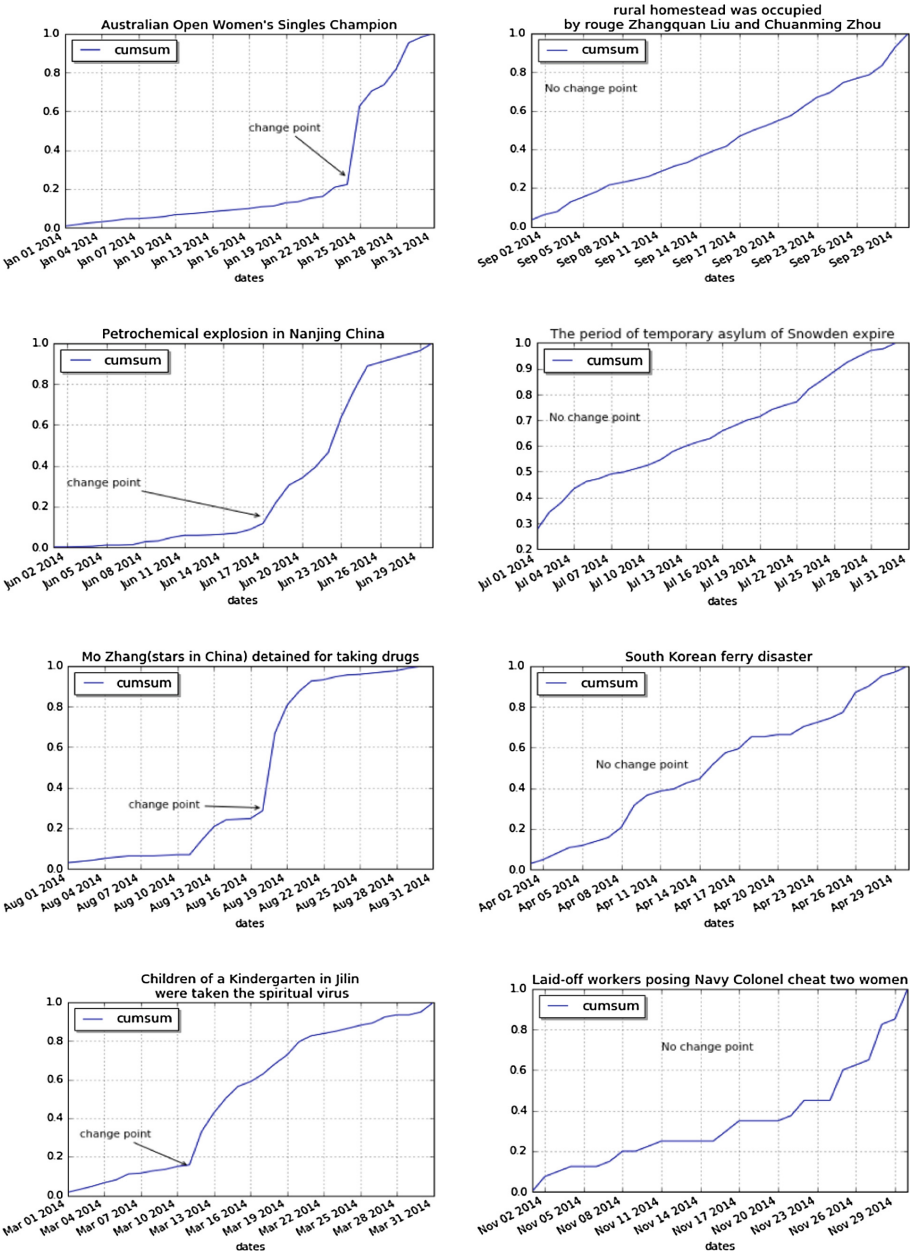

Fig. 5. Cumsum chart of events.

7 Conclusions

In this paper we proposed an efficient method to extract events from social media texts streams as well as a robust algorithm to identify hot events. In this method, the major contribution is listed as follows, first we considered the importance of source of doucuments when selecting keywords. Besides, the KeyGraph we built is an weighted graph which may capture the influence information of one keyword on another. It will improve the accuracy of community detection. Last but not least, we provide an efficient algorithm to detect the hot event. In the future work, early hot events detection is our main work.

References

1. Cataldi, M., Schifanella, C., Candan, K.S., Sapino, M.L., Di Caro, L.: Cosena: a context-based search and navigation system. In: Proceedings of International Conference on Management of Emergent Digital EcoSystems, p. 33. ACM (2009)
2. Farkas, I., Ábel, D., Palla, G., Vicsek, T.: Weighted network modules. New J. Phys. **9**(6), 180 (2007)
3. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text classification without negative examples revisit. IEEE Trans. Knowl. Data Eng. **18**(1), 6–20 (2006)
4. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 297–304. ACM (2004)
5. Li, Z., Wang, B., Li, M., Ma, W.-Y.: A probabilistic model for retrospective news event detection. In: Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 106–113. ACM (2005)
6. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web (1999)
7. Ramos, J.: Using TF-IDF to determine word relevance in document queries. In: Proceedings of 1st Instructional Conference on Machine Learning (2003)
8. Taylor, W.A.: Change-point analysis: a powerful new tool for detecting changes (2000). Preprint http://www.variation.com/cpa/tech/changepoint.html
9. Xie, W., Zhu, F., Jiang, J., Lim, E.-P., Wang, K.: Topicsketch: real-time bursty topic detection from Twitter. In: 2013 IEEE 13th International Conference on Data Mining, pp. 837–846. IEEE (2013)
10. Yang, Y., Ault, T., Pierce, T., Lattimer, C.W.: Improving text categorization methods for event tracking. In: Proceedings of 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 65–72. ACM (2000)
11. Yin, H., Cui, B., Lu, H., Huang, Y., Yao, J.: A unified model for stable and temporal topic detection from social media data. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), pp. 661–672. IEEE (2013)