Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 6, June 2018, pp. 1130-1139
DOI: 10.23940/ijpe.18.06.p4.11301139

What Is the Whole Development Process? Subevent Detection using Micro Index and Local Clustering

Hua Zhao[a,*], Qingtian Zeng[b], Yuqiang Zhang[a], and Weiyi Meng[c]

[a]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China*
[b]*College of Electronic, Communication and Physics, Shandong University of Science and Technology, Qingdao, 266590, China*
[c]*Computer Science Department, State Universtiy of New York at Binghamton, Binghamton, 13902, USA*

Abstract

Users can easily obtain a massive amount of news stories related to an event. But, often the obtained stories are fragmented and can only reflect certain aspects of the event. Detecting the subevents automatically is important for users to understand the whole development process of the event. Motivated by the co-evolution between the event and the opinions about it, we firstly propose to adopt Micro Index and give a dynamic time window construction method based on the recognition of the peaks of the Micro Index Curve. Secondly, we propose a two-stage subevent detection method based on local clustering and classification. And finally, we introduce the news stories about "Luo Yixiao Event" and "Shandong Illegal Vaccines Event", which are two recent hot events, and use them to evaluate the proposed methods. It is found that the proposed methods are successful.

Keywords: event; subevent detection; microIndex; time window; local clustering

(Submitted on March 10, 2018; Revised on April 26, 2018; Accepted on May 25, 2018)

1. Introduction

Nowadays, many events are reported, discussed and forwarded on social platforms, such as network news media, microblog, WeChat, blog, and so on, which bring out big event data [15]. How to mine the useful information from these massive and fragmented big data has become a new research topic [3,8]. Users can retrieve news stories about an event with the help of Internet tools, such as a Search Engine (SE), a Personalized Recommendation (PR) system, and so on. But these retrieved news stories could just show certain aspects of the event, not the whole development process, which makes users feel like "blind men touching an elephant". For users who want to know the complete process of an event [5], SEs and PR systems are inadequate. Some techniques, such as Topic Detection and Tracking (TDT), can organize the stories according to the events with a two-layer structure, in which all the stories related to the same event will be clustered together. In other words, TDT is incapable of presenting the main subevents of an event, so it can't be used to construct the development process of an event.

In view of the limitations discussed above, we propose to carry out our works with the following motivations:

(1) Analyzing the event big data automatically and detecting the subevents of a certain event are very helpful for the users to see the big milestones of the event.

(2) In the era of Web2.0, many events are discussed by users on online social platforms, which bring out Internet PublicOpinion (IPO). There exists co-evolution between the event and its related IPO: a new subevent will bring a new IPO, whereas when an IPO rises to a certain extent, the event actors will take some actions (which are considered as subevents in this paper). So, we believe that the changing trend of an IPO can be useful to subevents detection, which is proven by our experimental results.

* Corresponding author.
E-mail address: huamolin@163.com

The main contributions of this paper are listed as follows:

(1) We propose to adopt the changing trend of IPO to help subevent detection, where the trend is indicated by SinaMicro Index (MI) (http://data.weibo.com/index). To the best of our knowledge, this is the first work to use MI in subevent detection.

(2) We develop a dynamic construction method for time window based on the recognition of the MI curve peaks. Based on such recognition, the lifecycle of an event can be divided into several time windows, which are further classified into two types: key time window and rest time window.

(3) We propose a two-stage subevent detection algorithm based on local clustering and classification, where the Single-pass cluster algorithm and K-Nearest Neighbor (KNN) classification algorithm are adopted, respectively.

(4) By collecting news stories about "Luo Yixiao Event" (*Luo* for short) and "Shandong Illegal Vaccines Event" (*Vaccines* for short), we carry out experiments to evaluate the proposed methods.

2. Background and problem statements

2.1. Background

We firstly present several definitions, which are important for us to describe our subevent detection methods.

(1) News Story: A news story is a news article delivering some information to users. In our work, a news story is about a subevent. It is assumed that each story describes a single (but unnecessarily unique) subevent.

(2) Event: An event is something that happens at some specific time, and often at some specific place. In this paper, we don't restrict the event to happen "at some specific place", because some events actually do not happen at any specific places. For example, *Luo* attracts the interests of WeChat users but can't be assigned a specific place. It contains several subevents during the life cycle of the event.

(3) Subevent: A subevent of an event is a measure or an action taken by the event actors (maybe a person or an organization). During the development of the event, involved actors or organizations will take some actions to influence the event. For example, within Luo, the subevent may be "Luoer publish "Luo Yixiao, stop"", "Shenzhen Children's Hospital announces the medical cost", and so on. Our objective is to detect these important subevents.

(4) MicroIndex: MicroIndex is a statistical index about a certain hot event, which is launched by the Sina micro-blog. MicroIndex can reflect the influence of the event. Sina micro-blog has become a leader in information dissemination due to its timely news releasing, convenient interaction, and high news spreading speed. According to a report (http://www.useit.com.cn/thread-14392-1-1.html), as of Oct. 2016, there are 297 million active users in the Sina Micro-blog. Therefore, we believe that Micro Index can sufficiently reflect the opinion trend of an event.

2.2. Problem Statement

The objective of this paper is to develop an automatic method that can detect the important subevents of an event, which can help users capture the main subevents and understand the whole process of the events.

Consider the event *Luo*. When we use "Luo Yixiao" as the keyword to search Baidu and Google, we will have the following search results: Baidu returns 19,900,000 results and Google returns 3,400,000 results. Faced with the massive number of results, users can only view a small number of the (top) results and know some fragments of this event. As a result, they are likely to have a hard time in understanding how the event got started, what is the process of the event, and what the results of the event are. In other words, they probably can't know the full development process of the event without help.

Table 1 lists the subevents of *Luo* in time order. There are a total of 8 important subevents. From the list, users can understand what subevents have happened according to the time order and know how the event developed.

Table 1. Important Subevents of *Luo*

Subevents	Subevent Descriptions	Number of Stories
e1	Luo Yixiao catches a disease, the rewards in WeChat breakthrough the upper limit	137
e2	Users have doubt about the money, the sentiments reversed	894
e3	Luo Er responds to the doubt and explains the cost and the house	706
e4	Hospital publishes the medical cost and the reimbursement proportion	185
e5	Liu Xiafeng responds to the payment, and would like to set up the aid fund	414
e6	Civil affairs bureau investigates and appeals that the donations be handled by following the law	162
e7	After communication, money are finally returned to the donors	372
e8	Luo Yixiao died, and her parents would like to donate her body	114

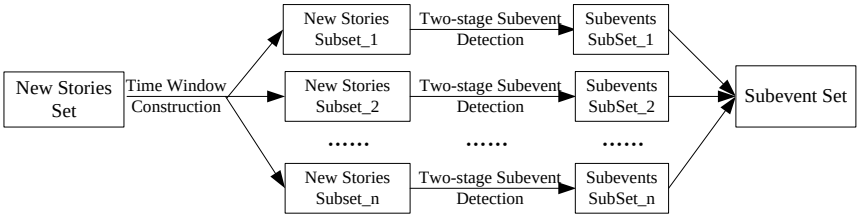


Figure 1. Architecture of Our Research

Figure 1 presents the architecture of our research. The input is news stories, and the output is the subevents. There are two main tasks, which are time window construction and two-stage subevent detection, and they will be introduced in Section 4.1 and Section 4.4, respectively.

3. Related works

Subevent detection is related to TDT. As a research topic about the organization of information based on the events, TDT has been an active research topic for many years [12]. Many techniques have been proposed for TDT. In order to extract the topics appearing in microblogs, J.Li et al. [6] organized the posts into a tree based on replies and forwards. M. He et al. [4] adopted a momentum model to detect bursty topics. Y. Liang et al. [7] detected the microblog hot events based on the distribution of emotion. L. M. Zhang et al. [16] adopted emoticons to detect online bursty events. Q. M. Diaoet al. [2] combined temporal information and user relationships when detecting the microblog bursty topics, which resulted in a better performance.

TDT generally focuses on how to cluster or classify news stories into different events, where news stories are organized into a flat hierarchical structure. From such a structure, users are not able to get a clear picture about the process of the event. Subevent detection can be thought as a further work beyond TDT.

Our research is also related to event extraction, which is a task of both the Message Understanding Conference and the Automatic Content Extractionand Text Analysis Conference. Many methods have been developed to extract events and they can be divided into two types: machine learning-based methods and rule-based methods. For example, D. Y. Zhou et al. [18] proposed a Bayesian-based method to extract Twitter events. Y. B. Chen et al. [1] made use of a Convolution Neural Networks in event extraction. D. Y. Zhou et al. [19] explored a kind of probabilistic model in event extraction. U. Shyam et al. [13] regarded event extraction as a classification problem and used Support Vector Machine (SVM) to extract the events. A. V. Marco et al. [10] explored a rule-based method, and proposed a domain independent event extraction method. L. Sha et al. [11] used regular expressions when extracting the events. Z. Zhu et al. [20] carried out extracting research on bilingual corpus.

Our subevent detection is different from the event extraction. Event extraction aims at extracting events from a sentence or a phrase, which focuses on a change of an action or a state. Our subevent detection emphasizes the whole process of an event, with the goal to construct the global view of the event and to help users see the big milestones of the event.

4. Proposed Methods

4.1. Dynamic Time Window Construction based on MicroIndex

4.1.1. Why we need time window?

The idea of a time window is to divide the news stories being analyzed into different groups, the motivations of which are listed as follows:

(1) The performance would be unsatisfactory when applying the clustering algorithms directly to the big event data [9]. By constructing time window, big data will be divided into multiple groups with smaller sizes. We can carry out local clustering within each group, and at the same time, different time windows may employ different clustering algorithms and may be clustered in parallel, which would speed up the analysis greatly.

(2) By analyzing the contents of the news stories, we observed that later stories usually cite the contents reported in the earlier ones, which may cause problems for the clustering of the later stories. This is because they will have overlapping content, which results in higher similarity than the actual similarity and leads to wrong clustering. Constructing time windows and carrying out local clustering within each time window will reduce this kind of problem.

4.1.2. Construction method for the time window

The time windows are usually defined as a static time interval [14], maybe one day or several hours, maybe a fixed number of news stories, and so on. Taking the co-evolution between the event and the online opinion into account, we propose a dynamic time window construction method based on the recognition of the peaks of the MicroIndex Curve (*MIC*).

From the *MIC* of *Luo* between Nov. 25, 2016 and December 30, 2016 (*http://data.weibo.com/index/hotword?wid=1061611300000145741&wname=%E7%BD%97%E4%B8%80%E7%AC%91*), we can observe clearly the time periods during which the event gets widespread attention. Such time periods are called the Key Time Window (*KTW*). For easy reference, we call the time period between two adjacent *KTW*s as a Rest Time Window (*RTW*). So, the life cycle of an event could be represented by a time window set $TW = \{ktw_1, ktw_2, \dots, ktw_m, rtw_1, rtw_2, \dots, rtw_n\}$, where each ktw_i $(1 \le i \le m)$ is a *KTW*, and each rtw_i $(1 \le i \le n)$ is a *RTW*. Table 2 shows the *KTW*s of *Luo*.

Table 2. The Time Windows of *Luo*

Key Time Window	Subevent
11.25-11.28	Luo Yixiao diagnosed withleukemia. The post of Luo Er spammed.
11.29-11.30	The donations in WeChat broke through the upper limit, users cast doubt about it. Hospital and Civil Affairs Bureau investigated
12.01-12.06	Donations returned. Luo Er responded.
12.14-12.23	Luo Er published posts again. Luo Yixiao entered ICU.
12.24-12.31	Luo Yixiao died, and her parents wanted to donate her organs.

Several issues needed to be addressed when constructing the time windows based on the *MIC*:

(1) The width of the time window cannot be pre-determined, because different subevents would have different durations.

(2) Our objective is to detect important subevents, so we need to get the time windows with significant peaks, not all the peaks.

(3) The peak threshold should be adjusted according to the change trend of the *MIC*. Smoother curves should have a lower threshold. But, when the curve fluctuation is relatively large, or the duration of the event is long, the threshold should be higher so the system can effectively find the important subevents.

Therefore, we propose to detect the curve peaks based on the signal detection algorithm z-score ($x_{z-score} = (x_i - \bar{x})/s$, where $\bar{x}$ is the mean of $\bar{x}$, s is the standard deviation), and then construct the time windows based on these detected peaks. Algorithm 1 shows the process. In the algorithm, we use *lag* to define the window size for comparison, and adopt the impact factor *influence*, which can help adjust the influence of the new value to the mean and variance. *threshold* is used to adjust the value of the z-score. Lines 1-4 perform the initialization. Lines 5-21 traverse all the subsequent micro index values and compute the z-score in turn. The time windows with the value 1 are the windows we are interested in.

Algorithm 1: Dynamic construction of time window based on the micro index curve

Input: value vector of micro index (y), slide window (*lag*), threshold (*threshold*), impact factor (*influence*)
Output: new values fitted using [-1, 0, 1]

1. Initialize $signals = [0,0, \dots,0]$, $signals$ is an array, the length of which is $len(y)$;
2. Get the component of the vector y based on the slide window lag, and let $filteredY = y[1: lag]$;
3. Create the dictionary $stdFilter$ and $avgFilter$ to record the variance and mean, respectively;
4. Let $avgFilter[lag] = mean(y[1: lag])$, $stdFilter[lag] = std(y[1: lag])$, where $mean()$ and $std()$ are used to

compute the mean and the variance of the data, respectively;
5. **for** $i \in [lag+1, \ldots, len(y)]$ **do**
6. **if** $abs(y[i] - avgFilter[i-1] > threshold \times stdFilter[i-1])$ **then**
7. **if** $y[i] > avgFilter[i-1]$ **then**
8. $signals[i] = 1$;
9. **else**
10. $signals[i] = -1$;
11. **end if**
12. $filteredY[i] = influence \times y[i] + (1 - influence) \times filteredY[i-1]$;
13. $avgFilter[i] = mean(filteredY[(i-lag):i])$;
14. $stdFilter[i] = std(filteredY[(i-lag):i])$;
15. **else**
16. $signals[i] = 0$;
17. $filteredY[i] = y[i]$
18. $avgFilter[i] = mean(filteredY[i-lag:i])$;
19. $stdFilter[i] = std(filteredY[i-lag:i])$;
20. **end if**
21. **end for**
22. **return** $signals$

4.2. News Story Model

We use the Vector Space Model (VSM) to represent the stories and the model consists of two parts: story content based VSM (C-VSM) and Named Entity based VSM (NE-VSM). Pre-processing is adopted before creating the content based VSM. Firstly, we use NLPIR (http://ictclas.nlpir.org/) for Chinese word segment and part of speech tagging, and we keep only the nouns, verbs, gerunds and adjectives. Secondly, stop words are removed.

News stories have two important characteristics. On one hand, story structure is like an inverted pyramid, where the important contents usually appear early in a story. Thus, when we select the features for the VSM, we only use the words in the first ε portion of a story. In our experiments, ε =1/3. On the other hand, the story title usually expresses the main topic of the story, which motivates us to increase the weight of the features appearing in the title. Accordingly, we use the following Equation (1) to compute the weights of the features:

$$w_{ik} = (\omega \times tf_t^k(t_i) + tf_c^k(t_i)) \times \frac{N}{n_i} \tag{1}$$

Where w_{ik} is the weight of the feature t_i in the story d_k, $tf_t^k(t_i)$ is the word frequency of t_i in the title of d_k, $tf_c^k(t_i)$ is the word frequency of t_i in the content of d_k, ω is an adjustable parameter, N is the number of the stories in the data set, and n_i is the number of stories that contain t_i.

There are usually some Named Entities (NE) in the news story, which may be the names of the event actors or the names of the related organizations. It is assumed that these NEs will be helpful to the accurate similarity computation between the news stories. So, we firstly use NLPIR to recognize NEs within the news stories and then create the *NE-VSM*, where the recognized NEs are the features and their word frequencies are used to compute their weights.

4.3. Similarity Computation Method

Once we get the story's *C-VSM* and *NE-VSM*, we use the following Equation (2) to compute the similarity (*Sim* (d_i, d_j)) between two stories (d_i and d_j):

$$Sim(d_i, d_j) = \frac{1}{3}\left[\alpha Cos(C_i, C_j) + \beta Jaccard(E_i, E_j) + \gamma Time(d_i, d_j)\right] \tag{2}$$

$$Cos(C_i, C_j) = \frac{\sum_{k=1}^{n} w_{ki} \times w_{kj}}{\sqrt{\sum_{k=1}^{n} w_{ki}^2} \times \sqrt{\sum_{k=1}^{n} w_{kj}^2}} \tag{3}$$

$$Jaccard(E_i, E_j) = \frac{|Ent_i \cap Ent_j|}{|Ent_i \cup Ent_j|} \tag{4}$$

$$Time(d_i, d_j) = e^{-\left[\frac{|Pub(d_i)-Pub(d_j)|}{T}\right]} \quad (5)$$

Where C_i and C_j are the *C-VSM* of d_i and d_j, respectively; E_i and E_j are the *NE-VSM* of d_i and d_j, respectively; $Cos(C_i, C_j)$ is the Cosine similarity computed by Equation (3); w_{ki} is the weight computed by Equation (1); $Jaccard(E_i, E_j)$is the Jaccard similarity computed by Equation (4); Ent_i and Ent_j are named entity sets from $d_i$and d_j, respectively; $Time(d_i, d_j)$ is computed by Equation (5), where $Pub(d_i)$ is the publish time of d_i, T is the duration of the event. α, β, γ are adjustable parameters.

4.4. Two-stageSubevent Detection based on local clustering and KNN

We propose a two-phase subevent detection algorithm based on local clustering and classification, which is shown in Algorithm 2. Lines 3-6 cluster the news stories within ktw_i $(1 \le i \le m)$ based on the Single-Pass method [17], where formula (1) is adopted to compute similarity and the threshold is Th_1. *Sub* may include one or more subevents. Similarly, lines 7-10 cluster the news stories within rtw_i $(1 \le i \le n)$, which will return a temp subevent set *R*. Lines 11-19 merge the subevents in *R* to *SubEv* based on the idea of KNN (K=1). When merging, if *MaxSim* is not larger than Th_2, r_i will be regarded as redundant and will be discarded. The algorithm returns *SubEv*.

Algorithm2 Subevent detection based on local clustering and KNN

Input: Story set $D = \{d_1, d_2, \dots, d_u\}$, cluster threshold Th_1, classification threshold Th_2
Output: Subevent set $SubEv = \{Sub_1, Sub_2, \dots, Sub_v\}$

1. Sort all the stories in *D* by time;
2. Create the time window $TW = \{ktw_1, ktw_2, \dots, ktw_m, rtw_1, rtw_2, \dots, rtw_n\}$ based on the Algorithm1;
3. **for** $i = 1$ to m **do**
4. $Sub = \text{Single} - \text{pass}(ktw_i, Th_1)$;
5. Add Sub to $SubEv$;
6. **end for**
7. **for** $i = 1$ to n **do**
8. $r = \text{Single} - \text{pass}(rtw_i, Th_1)$;
9. Add r to temp set R;
10. **end for**
11. **for** $r_i \in R$ **do**
12. **for** $Sub_j \in SubEv$ **do**
13. Compute the similarity between r_i and Sub_j;
14. end for
15. Suppose $MaxSim$ as the maximum similarity between r_i and all the Sub_jin $SubEv$;
16. **if** $MaxSim > Th_2$ **then**
17. Add r_i to Sub_j which achieves the maximum similarity;
18. **end if**
19. **end for**
20. **return** $SubEv$;

5. Experimental Results and Analysis

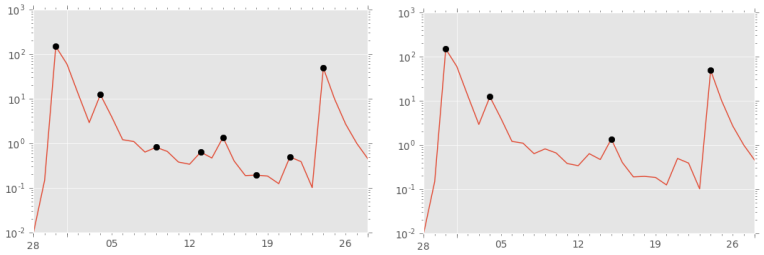
5.1. Corpus and Evaluation Method

The news stories related to *Luo* and *Vaccines* are collected by automatic crawling. We collect a total of 10,307 news stories. After filtering, we finally keep 2,984 stories for *Luo* (from 2016.11.25 to 12.25) and 5,616 stories for *Vaccines* (from 2016.3.15 to 5.1). The content, URL, title and publication time of each story are extracted.

After the data are collected, the subevents of these two events are manually identified by two people. As an example, Table 1 (in Section 2.2) shows the subevents of *Luo*. F-Measure is adopted to evaluate the proposed methods.

5.2. Experimental Results of Time Window Construction

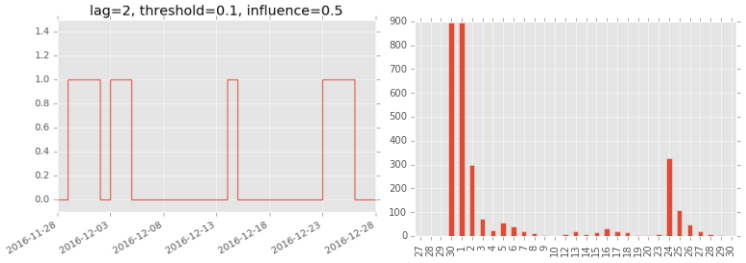
Taking *Luo* as an example, we first give the time window construction results. To make it easier to see the changes, we carry out a logarithmic transformation (provided by Python) to the original values in the original *MIC* of *Luo*, which results in the curve showed in Figure 2(a), from which we can see the change trend more clearly, and we mark all the peaks that satisfy $y_{x-1}<y_x< y_{x+1}$, where y is the logarithmic value, and x is the date. In order to detect the important subevents, we modify the restrictions from $y_{x-1}<y_x< y_{x+1}$ to $y_x–y_{x-1}>\theta$ and $y_x–y_{x+1}>\theta$, where θ is a pre-defined threshold and is set to 0.5 in our experiments. Then, we get Figure 2(b).



(a) (b)

Figure 2. Peaks of LuoMicro Index Curve

The curve showed in Figure 2(b) is converted to Figure 3(a) based on Algorithm 1. Continuous intervals with the value of 1 are the *KTWs*. Figure 3(b) is the timeline of the news stories from the original collected corpus, where the *x* axis represents the story publication time, and the *y* axis represents the user's attention to the stories (i.e., number of news articles). From Figure 3(a) and (b), we can see that there is a good match between the recognized *KTWs* and the user's attentions. This shows that our proposed dynamic construction algorithm for time windows based on Micro Index is effective.



(a) (b)

Figure 3. Curve fitting of time window and timeline

5.3. Experimental Results of Subevent Detection

We first carry out three groups of experiments to test the performance of our proposed subevent detection method; the results are reported in Table 3 and Table 4. The parameter settings are: α=0.8, β=0.1, γ=0.1, ω=5, Th_1=0.3, Th_2=0.45. *DWC* represents our proposed method, where the subevent detection is based on the dynamic construction of the time windows. *BC* represents the subevent detection based on the basic Single-Pass clustering where no time window is constructed. *SWC* represents the subevent detection based on the static time window.

Table 3. Subevent Detection Resultsof*Luo*

Algorithms	Precision (%)	Recall (%)	F-Measure (%)
DWC	93.3202	81.3004	86.8966
BC	67.2615	74.7280	70.7984
SWC	75.2577	83.6230	79.2201

Table 4. Subevent Detection Resultsof*Vaccines*

Algorithms	Precision (%)	Recall (%)	F-Measure (%)
DWC	83.7889	71.3041	77.0440
BC	64.8037	58.2501	61.3524
SWC	71.7113	65.1532	68.2751

From the results, we can see that *DWC* gets the best F-measure and significantly outperforms the two baselines. Both the F-measures of *DWC* and *SWC* are higher than that of *BC*, which means that creating time window is useful to subevent

detection. This is because the time windows (dynamic or static) can allow the local clustering, which can reduce the influence of the content overlap between the subsequent stories and the previous stories. From the comparison between the F-measure of *DWC* and that of *SWC*, we can see that dynamic time windows outperform static time windows, which shows that the co-evolution is helpful for subevent detection and making the best use of users' attention trend can be effective in constructing more accurate time windows.

Table 5 shows the subevent detection results, where Clusters are cluster numbers returned by our method, Subevents are the subevents manually annotated, and Feature Wordsare the top features.

Table 5. Subevent Detection Results of *Luo*

Clusters	Subevents	Feature Words
7	e1	function, upper limit, money amount, job, breakthrough, discuss, cause, burst, patient, fundraising, thanks
2	e2	money, resort to, reversion, story, make use of, family, truth, investigate, spread, fact, spam
8	e3	Liu Xiafeng, P2P observation, house, doubt, response, reimbursement, patient's condition, public number, medical fees, corporation, cost
4	e4	children, pay, children's hospital, hospital, child, patient's condition, proportion, infection, sum, medical care, notification, average
3	e5	Liu Xiafeng, declare, money amount, funds, help, payment, civil affairs bureau, affect, medical fees, receive, part, promise
6	e6	platform, charity law, donate, law, public welfare, organization, resort to, supervision, bilk, specification, control, intervention, forbid
1	e7	praise, funds, user, return, system, thanks, love, team, communication, platform, pay, record
5	e8	daughter, interview, parents, die, scolding, donate, body, children's hospital, organ

Figure 4 provides a visualization of the results by Inter-topic Distance Map, where each circle represents a detected subevent, the size of the circle indicates the number of the news stories about this subevent, and the distance between two circles represents the difference between the two subevents. From the visualization, we can see that Cluster 3 and Cluster 8 are very similar. From Table 1, we know that Cluster 8 (corresponding to e_3) is about "*Liu Xiafeng responds to the payment, and would like to set up the aid fund*" and Cluster 3 (Corresponding to e_5) is about "*Luo Er responds to the doubt and explains the cost and the house*". The contents of these two subevents are very similar, but our subevent detection method can distinguish them successfully, which verifies that our two-stage subevent detection method is very effective.

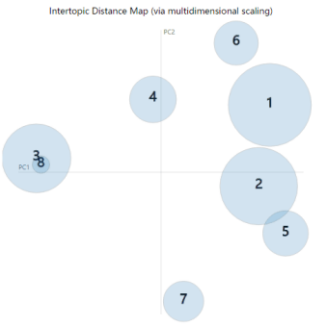


Figure 4. Inter-topic distance map of the cluster results about *Luo*

6. Conclusions

Detecting subevents automatically from the big event data is helpful for users to understand the full development process of events. In order to detect the subevents of an event quickly and effectively, in this paper, we firstly introduced utilizing MicroIndex and proposed a dynamic construction method for the time windows based on the detection of the peaks on the *MIC*. Secondly, we proposed a two-stage subevent detection method based on local clustering (within time window) and classification. Finally, we collected the news stories about *Luo* and *Vaccines*, which are two recent hot events, and used these data as the evaluation corpus. Experimental results showed that the proposed methods are highly effective in subevent detection.

Acknowledgements

This work is supported by the China NSFC program (No. 61602278); the Social Science and Humanity on Young Fund of the Ministry of Education (No. 16YJCZH154, No. 16YJCZH012); the SDUST Research Fund (No. 2015TDJH102); CAS

Key Lab of Network Data Science and Technology Open Fund Project (No. CASNDST201706); International Cooperation and Training Program for Outstanding Young Teachers in Shandong Colleges and Universities.

References

1. Y. B. Chen, L. H. Xu, K. Liu, D. J. Zeng, and J. Zhao, "Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks," *in Proceedings of International Joint Conference on Natural Language Processing*, pp. 167-176, Beijing, China, July 2015
2. Q. M. Diao, J. Jiang, F. D. Zhu, and E. P. Lim, "Finding Bursty Topics from Microblogs," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 536-544, Jeju, Korea, July 2012
3. T. Ge, W. Z. Pei, H. Ji, S. J. Li, B. B. Chang, and Z. F. Sui, "Bring You to the Past: Automatic Generation of Topically Relevant Event Chronicles," *in Proceedings of International Joint Conference on Natural Language Processing*, pp. 575-585, Beijing, China, July 2015
4. M. He, P. Du, J. Zhang, Y. Liu, and X. Q. Cheng, "Microblog Bursty Topic Detection Method based on Momentum Model," *Journal of Computer Research and Development*, vol. 52, no. 5, pp. 1022-1028, 2015
5. F. H. Li, D. Q. Zheng, and T. J. Zhao, "Dynamic Incremental Analysis of Sub-Topic Evolution," *Journal of Computer Research and Development*, vol. 52, no. 11, pp. 2441-2450, 2015
6. J. Li, M. Liao, W. Gao, Y. L. He, and K. F. Wong, "Topic Extraction from Microblog Posts Using Conversation Structures," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 2114-2123, Berlin, Germany, August 2016
7. Y. Liang, Y. Lin, and H. F. Lin, "Micro-blog Hot Events Detection based on Emotion Distribution," *Journal of Chinese Information Processing*, vol. 26, no. 1, pp. 84-91, 2012
8. H. L. Lin, Y. Z. Wang, Y. T. Jia, P. Zhang, and W. P. Wang, "Network Big Data Oriented Knowledge Fusion Methods: a Survey," *Chinese Journal of Computers*, vol. 40, no. 1, pp. 11-27, 2017
9. R. Lu, L. Xiang, M. R. Liu, and Q. Yang, "Discovering News Topics from Microblogs Based on Hidden Topics Analysis and Text Clustering," *Pattern Recognition and Artificial Intelligence*, vol. 25, no. 3, pp. 382-387, 2012
10. A. V. Marco, H. P. Gus, S. Mihai, and H. Thomas, "a Domain-Independent Rule-based Framework for Event Extraction," *in Proceedings of International Joint Conference on Natural Language Processing*, pp. 127-132, Beijing, China, July 2015
11. L. Sha, J. Liu, C. Y. Lin, S. J. Li, B. B. Chang, and Z. F. Sui, "RBPB: Regularization-Based Pattern Balancing Method for Event Extraction," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 1224-1234, Berlin, Germany, August 2016
12. B. Shi, W. Lam, L. D. Bing, and Y. Q. Xu, "Detecting Common Discussion Topics across Culture from News Reader Comments," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 676–685, Berlin, Germany, August 2016
13. U. Shyam, C. Christos, and R. Dan, "Making the News: Identifying Noteworthy Events in News Articles," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 1-7, Berlin, Germany, August 2016
14. S. Unankard, X. Li, M. A. Sharaf, "Emerging event detection in social networks with location sensitivity," *World Wide Web*, vol. 18, no. 5, pp. 1393–1417, 2015
15. Z. H. Wu, C. Liang, and C. L. Giles, "Storybase: Towards Building a Knowledge Base for News Events," *in Proceedings of International Joint Conference on Natural Language Processing*, pp. 133-138, Beijing, China, July 2015
16. L. M. Zhang, Y. Jia, B. Zhou, J. H. Zhao, and F. Hong, "Online Bursty Events Detection based on Emoticons," *Chinese Journal of Computers*, vol. 36, no. 8, pp. 1659-1667, 2013
17. H. Zhao, T. J. Zhao, H. Yu, and D. Q. Zheng, "Using Word Differentiation and Temporal Information in English Topic Detection," *Journal of computational information systems*, vol. 2, no. 4, pp. 1473-1480, 2006
18. D. Y. Zhou, L. Y. Chen, and Y. L. He, "a Simple Bayesian Modelling Approach to Event Extraction from Twitter," *in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 700-705, Baltimore, Maryland, USA, June 2014
19. D. Y. Zhou, T. M. Gao, and Y. L. He, "Jointly Event Extraction and Visualization on Twitter via Probabilistic Modelling," *in Proceedings of the 54nd Annual Meeting of the Association for Computational Linguistics*, pp. 269-278, Berlin, Germany, August 2016
20. Z. Zhu, S. S. Li, G. D. Zhou, and R. Xia, "Bilingual Event Extraction: a Case Study on Trigger Type Determination," *in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 842-847, Baltimore, Maryland, USA, June 2014

Hua Zhao received her BS degree in Computer Science and Technology from Liao Cheng University, Liao Cheng, China, in 2001, her MS degree and her PhD degree in Computer Application Technology from Harbin Institute of Technology (HIT), Harbin, China, in 2003 and 2008, respectively. She currently is an Associate Professor in the College of Computer Science and Engineering, Shandong University of Science and Technology, China. Her research interests include process mining and text mining.

Qingtian Zeng received his BS degree and MS degree in Computer Science from Shandong University of Science and Technology, Taian, China, in 1998 and 2001 respectively, and his PhD degree in Computer Software and Theory from the

Institute of Computer Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a professor at Shandong Univerity of Science and Technology, Qingdao, China. His research interests include Petri nets, process mining, and konwledge management.

Yuqiang Zhang received his BS degree in Computer Science from Shandong University of Science and Technology, Qingdao, China, in 2014. He is now a Master's student at the Shandong University of Science and Technology. His research is in event detection.

Weiyi Meng received his BS degree in Mathematics from Sichuan University, Sichuan, China, in 1982, and his MS and PhD degrees in Computer Science from the University of Illinois at Chicago, in 1988 and 1992, respectively. He currently is a professor in the department of Computer Science at the State Universtiy of New York at Binghamton, Binghamton, USA. His research interests include large-scale metasearch engines, web database integration, sentiment analysis, and Internet-based information retrieval.