EventSys: Tracking Event Evolution on Microblogging Platforms

Lin Mu[1], Peiquan Jin[1](✉), Lizhou Zheng[1], and En-Hong Chen[1,2]

[1] University of Science and Technology of China, Hefei, China
jpq@ustc.edu.cn
[2] Anhui Province Key Laboratory of Big Data Analysis and Application, Hefei, China

Abstract. In this paper, we demonstrate a prototype system named EventSys, which provides efficient monitoring services for detecting and tracking event evolution on microblogging platforms. The major features of EventSys are: (1) It describes the lifecycle of an event by a staged model, and provides effective algorithms for detecting the stages of an event. (2) It offers emotional analysis over the stages of an event, through which people are able to know the public emotional tendency over a specific event at different time. (3) It provides a novel event-type-driven method to extract event tuples, which forms the foundation for event evolution analysis. After a brief introduction to the architecture and key technologies of EventSys, we present a case study to demonstrate the working process of EventSys.

Keywords: Event evolution · Emotional evolution · Tracking Microblog · Detection

1 Introduction

Microblog platforms have been one of the major sources for new events detection and spreading. Motivated by the massive fresh information generated by microblog users, many works on event detection and analysis on microblogs have been conducted in recent years [1–4]. However, previous studies mainly focused on extracting structural tuples of events, e.g., extracting the 5W1H (who, where, when, what, whom, how) information [2]. In addition to event tuple extraction, some studies paid attention to the evolution analysis of events [5], but they cannot grasp the development process of events. On the other hand, an event usually has a developing process in the real world, i.e., from birth to death, which is similar to the lifecycle of people. The lifecycle information of an event is very useful in information mining and decision making. For example, company managers can make specific decisions according to the developing stage of the events related to products.

In this paper, we propose to extract the lifecycle of events from microblogs. Basically, the lifecycle of an event can be defined as a five-stage process including a budding stage, a developing stage, a peak stage, a recession stage, and a pacification stage. Although there are some previous studies focusing on event evolution [6], to the best of our knowledge, they are not able to extract the lifecycle of events.

J. Pei et al. (Eds.): DASFAA 2018, LNCS 10828, pp. 797–801, 2018.
https://doi.org/10.1007/978-3-319-91458-9_51

Particularly, we propose a prototype system called EventSys for detecting and analyzing the lifecycle of events from microblogs. The major features of EventSys are as follows:

(1) *Microblog Event Tuple Extraction and Semantic Element Extraction*. Given an event keyword, one problem is how to effectively extract the keyword-related events from the microblog set. We propose to incorporate event type into the event tuple extraction. Inspired by the studies in the news-report area that describe an event based on the news features [1], i.e., when, where, who, whom, what, and how, we consider to detect the news features of events from microblogs.
(2) *Microblog Event Evolution Stage Detection*. In order to grasp the lifecycle of events, we describe the lifecycle of events based on a five-stage model that consists of five stages: budding, development, peek, recession, and pacification.
(3) *Emotional Evolution Analysis*. The public emotional tendency to an event varies with time. Based on the extracted stages of an event, we develop a visual interface to monitor the public emotional evolution for each stage of specific events.

2 Architecture and Key Technologies of EventSys

Figure 1 shows the architecture of EventSys. The modules of EventSys include event tuple extraction, event tuple linking, event lifecycle detection, and emotional evolution analysis.

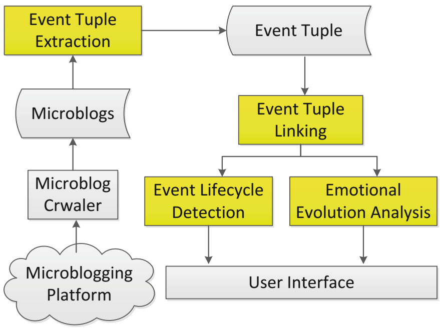

Fig. 1. Architecture of EventSys

Event Tuple Extraction. The extraction of event tuples is based on event type, which is defined as follows.

Definition 1. *Event Type. Given a collection of microblog post T which is obtained by one event query word, the event type is defined as a* quadruple $< p_l, p_n, p_o, p_t >$, *in which* p_l, p_n, p_o, p_t *represent the importance of location, person name, organization and time entity in the collection respectively, and* $p_l + p_n + p_o + p_t = 1$.

Given a microblog post collection, we represent it as a feature vector $\boldsymbol{x}$ and then employ the Multinomial Logistic Regression method to train the model [1]. The result $p_i = p\left(y = i|x^{(i)}, w\right)$ where $i = l, p, o$ and t for different named entity categories is used as the probabilistic distribution. Then, we use the quadruple $< p_l,\ p_n,\ p_o,\ p_t >$ to represent the event type, based on which we perform event-type-based clustering for microblogs by calculating the similarity among microblogs. We use the named entity probability distribution to adjust the similarity of the named entity of the microblog text to enhance the extraction effect, and finally get several events microblogging clusters. Each cluster of microblogging describes the same event. Next, for each cluster, we extract the 5W1H information [2], and finally get event tuples.

Event Tuple Linking. After extracting event tuples, we need to link the event tuples that describes the same event. This is mainly because an event will evolve with time. Given the microblogging data set at time t_i, we first get the set of event tuples, represented by eventTupleSet$_i$. For each event tuple in eventTupleSet$_i$, we calculate the similarity of the event tuple to previous events, find the most similar event, and link the event tuple to that event. If there is no similar event, we create a new event and add attach the event tuple to the newly created event.

Event Lifecycle Detection. Figure 2 shows the representation framework of an event. Each event has a unique ID and a set of event attributes. It also has a unique lifecycle that is five-bit structure indicating the current evolution process of the event. An event has a list of event tuples that are linked by the event tuple linking algorithm. All event tuples are arranged along the timeline and each tuple has an indicator describing what stage it belongs to.

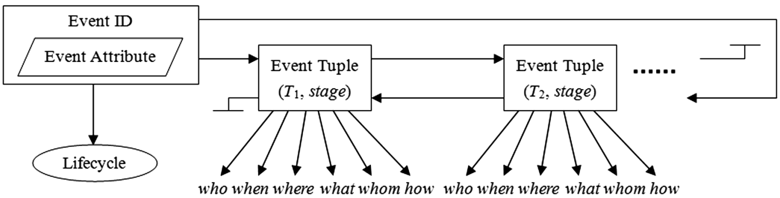

Fig. 2. Event representation

The key issue of event evolution analysis is to determine the right stage of an event. Sometimes we need to predict the stage of an event in future. In our system, we use the popularity of events to detect the event lifecycle. We define the popularity of an event in terms of the following features: (1) The forwarding number and commenting number as well as the total number of related microblog posts are used to measure the popularity of an event. (2) If users' emotional tendency towards an event changes dramatically, it implies that the evolutional stage of the event may change. (3) People cannot always focus on one specific event. When a new and interesting event happens, it will attract user attention and change the evolutional stage of current events.

(4) When the locations that are embedded in the event tuples change, it usually implies the change of the evolutional stage of the event.

2.1 Emotional Evolution Analysis

The public emotional tendency over a certain event will change with time. Thus, it is helpful to track the emotional evolution of events in decision making. In EventSys, we take three steps to extract the emotional evolution information of an event: (1) First, We extract the microblogging event tuples from each microblogging slices and extract the emotional tendencies of the event tuple based on a given sentiment dictionary. (2) Second, we map the event tuple to a developing stage of the event, as shown in the event representation framework in Fig. 2. (3) Finally, we compute the overall emotional polarity for each stage of an event based on the emotional tendencies of the event tuples linked within the stage. We use a weighted sum to aggregate the emotional tendencies of the event tuples in a stage, in which we put high weights for recent event tuples.

3 Demonstration

Figure 3 shows a screenshot of *EventSys*. Users are first required to select the time interval and event keywords as well as other parameters. The system will extract all events related to the selected event keywords. For example, Fig. 3 shows the output

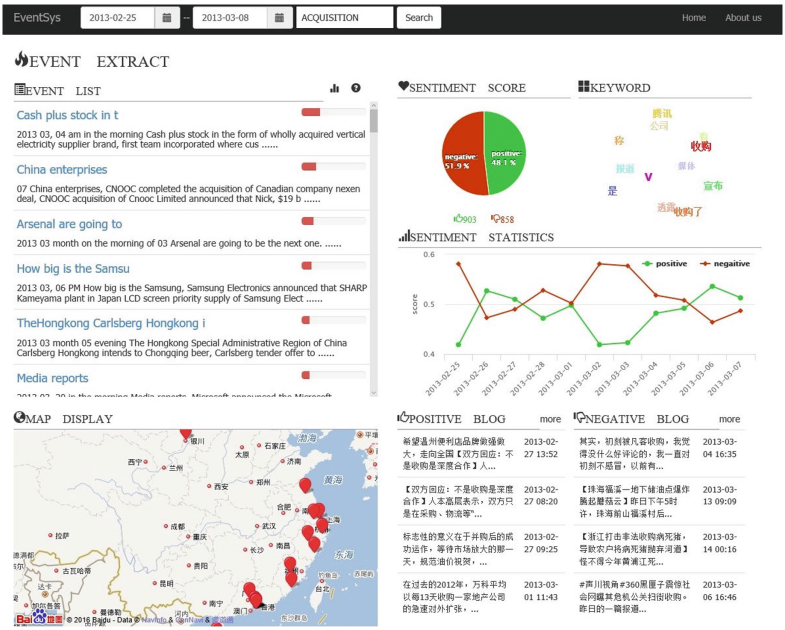

Fig. 3. Screenshot of EventSys

after inputting the event keyword "*acquisition*", Zone A shows a list of all the events associated with the keyword. The event is sorted by the number of related microblogs. In zone B, we can see that the acquisition events occurred mostly in the eastern coastal areas of China. The right part of Fig. 3 shows the emotional evolution of the event, where different kinds of emotional information are presented, including the static statistical emotion, the dynamic emotional tendency, and the supported microblog posts.

Acknowledgements. This work is supported by the National Science Foundation of China (61672479 and 71273010).

References

1. Jin, P., Mu, L., et al.: News feature extraction for events on social network platforms. In: WWW, pp. 69–78 (2017)
2. Zheng, L., Jin, P., Zhao, J., Yue, L.: A fine-grained approach for extracting events on microblogs. In: Decker, H., Lhotská, L., Link, S., Spies, M., Wagner, Roland R. (eds.) DEXA 2014. LNCS, vol. 8644, pp. 275–283. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10073-9_22
3. Sakaki, T., Okazaki, M., et al.: Earthquake shakes twitter users: real-time event detection by social sensors. In: WWW, pp. 851–860 (2010)
4. Zhao, J., Wang, X., et al.: Feature selection for event discovery in social media: a comparative study. Comput. Hum. Behav. **51**, 903–909 (2015)
5. Cai, H., Huang, Z., et al.: Indexing evolving events from tweet streams. In: ICDE, pp. 1538–1539 (2016)
6. Huang, J., Peng, M., et al.: A probabilistic method for emerging topic tracking in microblog stream. World Wide Web **20**(2), 325–350 (2017)