2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

# Semantic Social Event Evolution Discovering

Zihang Huang*, Lianzhong Liu*, Jiaofu Zhang*, Lihua Han†, Shuhai Wang†, Tongge Xu*, Yangyang Li‡, Yifeng Liu‡, Md Zakirul Alam Bhuiyan§
* School of Cyber Science and Technology, Beihang University, Beijing 100191, China
† School of Continuing Education, Shijiazhuang Railway University, Shijiazhuang 050043, China
‡ National Engineering Laboratory for Risk Perception and Prevention (NEL-RPP), China Academy of Electronics and Information Technology, Beijing 100041, China
§ Department of Computer and Information Sciences, Fordham University, NY 10458, USA
Email: {pppihf,lz_liu,xutg}@buaa.edu.cn, zhangliangzjf@qq.com, {hanlihua,wangshuhai}@stdu.edu.cn, {liyangyang,liuyifeng3}@cetc.com.cn, mbhuiyan3@fordham.edu

***Abstract*—People tend to express their opinions and emotions on social platforms, such as Weibo and Twitter. Although recent researches can discover topics and detect events from social messages, people pay more attention to the evolution analysis of events. However, existing methods only utilize the statistical characteristics of the text instead of semantic information. In this paper, we construct a 6-tuple to represent events to analyze the correlation between events from multiple perspectives. In order to mine the evolutionary relationship between events, we develop the method for measuring event similarity by concerning entities and texts. In terms of textual similarity measuring, we do a domain-adaptive incremental training task on the pre-trained NLP model to gain word embeddings for semantic information. Experiments on real social datasets show that our technique precedes the baseline technique.**

***Index Terms*—event evolution, similarity measure, social events, word embedding**

## I. Introduction

With the rapid development of the Internet, social platform plays an increasingly important role in human society. Represented by Weibo and Twitter, people record their lives, share their ideas, and comment on things on social platforms. Moreover, the popularity of mobile devices makes real-life events spread rapidly on social platforms, breaking the restrictions of time and space. Therefore, people can discover and analyze events from the social messages [1–4].

Real-life events are in dynamic change, which may lead to new events, split into several events, or die out. Event evolution analysis can be applied to event prediction [5], public opinion monitoring [1], cybersecurity [6], etc. To discover the development process of events, existing techniques evaluate the correlation between events [7]. Most of them focus on the structural and statistical characteristics of the text, so they are not quite fit for short-text social messages. In dealing with an extensive vocabulary and many long-tail words and low-frequency words, previous methods are unable to acquire event relationships that are interpretable and easily understood.

To address the above limitations, we take word embedding into account. Word embedding is a typical example of representation learning which is widely used in text classification [8–10], heterogeneous information network [11–13], graph neural network [9, 14–20] and urban traffic passenger flows prediction [21, 22]. Word embedding uses contextual information during training and integrates the features of surrounding words, thereby mapping words into a high-dimensional space containing semantic information. Recently, various models using large-scale corpus pre-training have greatly improved the quality of word embedding in specific downstream tasks, such as ELMo [23], BERT [24] and RoBERTa [25]. With the pre-trained word embedding, we introduce semantic information for event similarity measures in our method. The main contributions of this paper are listed as follows:

- We construct a 6-tuple to represent events, including time, location, participants, keywords, summary, and related posts.
- We develop an event similarity measure function to analyze event relationships by using pre-trained word embedding.
- Experiments on a manually annotated dataset demonstrate the effectiveness of our method compared with the baseline methods.

The rest of the paper is organized as follows. Firstly, Section II presents preliminary and problem definitions. Section III introduces our method in detail. Section IV shows experimental results. Finally, we conclude the paper in section V.

## II. Preliminaries and Problem Definition

In this section, we introduce some basic concepts, necessary notations and the problem formulation related to this research.

*Event.* While a variety of definitions of **event** have been suggested, this paper will use the definition suggested by [2] which regards an event as a set of correlated social messages that discuss the same real-world happening. Generally, events contain a variety of characteristics, so we use a 6-tuple to represent an event:

$$E =< t, loc, par, key, sum, m > \quad (1)$$

where $t$ is the timestamp of the event. $loc$ is the location mentioned in the event. $par$ is the participants involved in the event. $key$ and $sum$ are keywords and summaries that can briefly describe the event, respectively. $m$ is a set of social messages related to the event. In this paper, a social message can be included in at most one event to ensure that there is

978-0-7381-2646-3/21/$31.00 ©2021 IEEE

DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00171

no overlap between events. To illustrate 6-tuple more clearly, we take a recent hot event as an example in Table I, which is about Park Geun-hye's best friend Choi Soon Sil claiming to have been sexually harassed in the detention center.

TABLE I
AN EXAMPLE FOR 6-TUPLE

| | |
|---|---|
| $t$ | 2021/04/12 10:54 |
| $loc$ | South Korea |
| $par$ | Park Geun-hye, Choi Soon Sil, Director of Detention Center, Medical Section Chief |
| $key$ | Park Geun-hye, Choi Soon Sil, sexual harassment, Detention center |
| $sum$ | Park Geun-hye's best friend Choi Soon-sil claimed to have been sexually harassed in the detention center |
| $m$ | According to Yonhap News Agency, South Korean police reported on the 12th that Choi Soon-sil filed a complaint with the Grand Prosecutor's Office a few days ago, claiming that the director of the Cheongju Women's Detention Center and the head of the medical section were suspected of sexual harassment, negligence, and abuse of power. ... |

*Event evolution.* In real life, we say that one event evolves into another if two different events discuss the same topic. For example, $E_i \rightarrow E_j$ may denote $E_i$ is the cause of $E_j$, or $E_j$ is a further fermentation of $E_i$, or $E_i$ and $E_j$ are only related in content.

*Problem formulation.* This study aims to evaluate the relationship between events and finally discover the events' evolution paths. Given a set of events $\mathcal{E} = \{E_1, E_2, \ldots, E_N\}$, which are detected from social messages and arranged in chronological order, the evolutionary relationship between events is formalized as a upper triangular matrix $R_{N\times N}$. Under the assumption of $1 \leq i < j \leq N$, $R_{i,j} = 1$ indicate $E_i$ is the previous event of $E_j$, or $E_j$ evolved from $E_i$, $R_{i,j} = 0$ indicate the similarity between $E_i$ and $E_j$ is insufficient to establish an evolutionary relationship.

According to an evolutionary relationship matrix $R$, we can generate an event evolution graph $\mathcal{G}$, which should be a directed acyclic graph (DAG). In Fig 1, $E_1$ is the start event, evolving into two events $E_3$ and $E_5$. $E_5$ evolved from two events $E_1$ and $E_4$. $E_6$ is the end event.

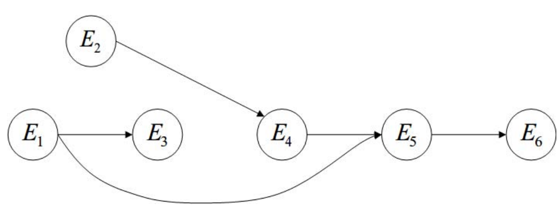


Fig. 1. An Example for Event Evolution Graph

## III. APPROACH

In this section, we introduce our approach for measuring similarity of events and generating event evolution graph with the combination of the inverted index.

### A. Event Detection

The generation of event evolution graph crucially depends on effective event detection. Some existing event detection methods achieve good results in both online [2] and offline [26] scenarios. The details of the event detection algorithm can be ignored here since our focus is on the event evolution algorithm in the following subsections.

We get several clusters of social messages through the event detection algorithm, where a cluster is an event. To characterize these events more comprehensively, we use the information in social messages and additional knowledge to construct a 6-tuple for every event. As far as $E_i$ is concerned, $t^i$ is the time when $E_i$ occurs. $E_i$ is usually accompanied by a timestamp when it is a component of outputs from event detection. However, it seems not appropriate to set the time stamp as $t^i$ because of delay in the case of offline. As a replacement, we extract real time (if any) from social messages, or we use the timestamp when users create the messages. In the same way, we gain named entities such as places, people, and organizations from tokenized texts with NLP tools. The places mentioned are set as $loc^i$, and the participants involved are set as $par^i$. Both of them may be empty, which is permitted in this paper. Some tweets/posts contain hashtags ("#XX"), which clarify themselves into different topics and play the role of archive aggregation. These messages have more probability of being searched and interacted. Therefore, if hashtags exist, they are set as $key^i$. We apply off-the-shelf tools to extract keywords for $key^i$, which is determined in the experiment setup. As $sum^i$, the sentence with the most keywords can be regarded as the core of the event, through which people can understand the whole event. The last part of the 6-tuple is $m^i = \{m^i_1, m^i_2, \ldots\}$, which is the social messages related to $E_i$.

### B. Event Similarity Measures

To discover the evolutionary relationship between events, similarity measurement is an essential part. According to 6-tuple, we construct an event similarity measure function for $E_i = < t^i, loc^i, par^i, key^i, sum^i, m^i >$ and $E_j = < t^j, loc^j, par^j, key^j, sum^j, m^j >$ as follows:

$$\begin{aligned} Sim(E_i,E_j) = {} & \alpha \cdot Sim_t(t^i,t^j) + \beta \cdot Sim_{loc}(loc^i,loc^j) + \\ & \gamma \cdot Sim_{par}(par^i,par^j) + \epsilon \cdot Sim_{key}(key^i,key^j) + \\ & \rho \cdot Sim_{sum}(sum^i,sum^j) + \omega \cdot Sim_m(m^i,m^j) \end{aligned} \tag{2}$$

which contains six independent similarity measures for each component in 6-tuple, and six corresponding weights. Meanwhile, we impose $\alpha + \beta + \gamma + \epsilon + \rho + \omega = 1$ as the limiting condition.

Intuitively, the correlation between two events with shorter time intervals is stronger. On the other hand, it is not easy to trace most events back to another event long ago directly. However, the rate of correlation decay varies with the increase of time difference, and we use an exponential function to measure the temporal similarity, which is defined as below.

$$Sim_t(t^i,t^j) = e^{-\frac{\left|t^i - t^j\right|}{T}} \tag{3}$$

where $T$ is the time distance between the earliest and the latest event.

Location and participant are two significant entities in event attributes, affecting the evolutionary relationship of two events to a great extent. The occurrence of events in the same places or involving the identical participants may lead to a strong correlation. We use Jaccard index to measure the similarity between two location sets:

$$Sim_{loc}(loc^i, loc^j) = \frac{|loc^i \cap loc^j|}{|loc^i \cup loc^j|} \tag{4}$$

The same is true of the participant sets:

$$Sim_{par}(par^i, par^j) = \frac{|par^i \cap par^j|}{|par^i \cup par^j|} \tag{5}$$

Keywords are the refinement and concentration of an event, while summaries give people the central idea of an event. They are obtained from the text of social messages. Therefore, an effective text similarity measurement method is the key to evaluate the relevance of events. Instead of counting word frequency, such as TF-IDF, which may lead to errors even if two pieces of text are similar in form, we take the semantics of text into account. Pre-trained language models using large-scale corpus capture the positional relationship between words and map words into fixed-length high-dimensional embedding, containing the interactive information of words. We employ cosine similarity to measure the similarity between embeddings. Given two vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, $n$ is the dimension of vectors, cosine similarity is defined as follows:

$$Sim_{cos}(x, y) = \frac{\sum_{k=1}^{n}(x_k \cdot y_k)}{\sqrt{\sum_{k=1}^{n} x_k^2} \cdot \sqrt{\sum_{k=1}^{n} y_k^2}} \tag{6}$$

On the basis of (6), we calculate the cosine similarity between each keyword in set $key^i$ and each keyword in set $key^j$, and then average them as follows:

$$Sim_{key}(key^i, key^j) = \frac{\sum_{w_i \in key^i} \sum_{w_j \in key^j} Sim_{cos}(w_i, w_j)}{|key^i| \cdot |key^j|} \tag{7}$$

where $w_i, w_j \in \mathbb{R}^n$ are the embeddings of keywords contained in $key^i$ and $key^j$ respectively.

We represent each sentence in the text to an embedding that has the same dimensions as the word embedding, and the value of each dimension is the average of the embedding of all words in the sentence on that dimension. Accordingly,the similarity of summaries is calculated by the following equation:

$$Sim_{sum}(sum^i, sum^j) = Sim_{cos}(emb_{sum^i}, emb_{sum^j}) \tag{8}$$

where $emb_{sum^i}, emb_{sum^j} \in \mathbb{R}^n$ are the embeddings of summaries contained in $sum^i$ and $sum^j$ respectively.

For the messages contained in the event, we calculate the average value of cosine similarity as for the keywords:

$$Sim_m(m^i, m^j) = \frac{\sum_{s_i \in m^i} \sum_{s_j \in m^j} Sim_{cos}(s_i, s_j)}{|m^i| \cdot |m^j|} \tag{9}$$

where $s_i, s_j \in \mathbb{R}^n$ are the embeddings of messages contained in $m^i$ and $m^j$ respectively. Too many elements in the collection may cause performance degradation, so we limit the size of $key$ and $m$.

Due to the enormous amount of corpus, another problem is that the vector space generated by the pre-trained model is too large, and the social message texts have a strong sparsity in this space. Hence, we extract part of the natural social messages as the training corpus and do domain-adaptive pretraining based on the pre-trained model.

### C. Event Evolution Graph Generation

We can generate the event evolution graph according to the defined event similarity measure function. Given a set of events that are processed into 6-tuples, we first arrange them in chronological order and then calculate the similarity between each pair of them by (2). We set a threshold $\delta$, and only when $Sim(E_i, E_j) > \delta$ is there an evolutionary relationship between $E_i$ and $E_j$. In this way, we get an matrix $R_{N\times N}$ when the amounts of events is $N$, and the element values in the matrix are as follows:

$$R_{i,j} = \begin{cases} 1, & Sim(E_i, E_j) > \delta \\ 0, & Sim(E_i, E_j) \leq \delta \end{cases} \tag{10}$$

Note that $R$ is an upper triangular matrix since $E_i$ occurs after $E_j$ when $i > j$.

However, there are many messages on social platforms, from which thousands of events may be detected. Traversing all the events will waste much time calculating the similarity between two unrelated events, particularly between message sets, which is pointless. To tackle this, we use an inverted index to reduce the amount of calculation, thereby improving efficiency.

**Algorithm 1:** Event Evolution Graph Generation

**Input:** Events set $\mathcal{E}$
**Output:** Graph $\mathcal{G}$
Initialize matrix $R_{N\times N}$ by zero;
Initialize graph $\mathcal{G}$ by event nodes;
Build inverted index $EI$;
**for** $E_i$ *in* $\mathcal{E}$ **do**
  Get events set $\mathcal{E}'$ related to and later then $E_i$ by searching $EI$;
  Arrange $\mathcal{E}'$ in chronological order;
  **for** $E_j$ *in* $\mathcal{E}'$ **do**
    Calculate $Sim(E_i, E_j)$;
    **if** $Sim(E_i, E_j) > \delta$ **then**
      $R_{i,j} = 1$;
      Add a directed edge from $E_i$ to $E_j$ in $\mathcal{G}$;
    **end**
  **end**
**end**
**return** $\mathcal{G}$

We assume that related events share at least one noun in their message sets. For event $E_i$, we randomly select ten

TABLE II

THE EVENTS LIST OF "JAPAN'S NUCLEAR WASTEWATER DISCHARGE INTO THE SEA"

| No. | Event Name | Num. | Time |
|---|---|---|---|
| $E_1$ | The Japanese government basically decided to discharge Fukushima nuclear wastewater into the sea | 126 | 04/09 13:38 |
| $E_2$ | German research shows that Japan's nuclear wastewater will pollute half of the Pacific Ocean in 57 days | 136 | 04/11 10:00 |
| $E_3$ | FMPRC responds to Japan's proposed decision to discharge nuclear wastewater into the sea | 54 | 04/12 15:19 |
| $E_4$ | Yoshihide Suga responds to Fukushima nuclear wastewater discharged into the sea | 67 | 04/12 18:34 |
| $E_5$ | The Japanese government has officially decided to discharge Fukushima nuclear wastewater into the sea | 197 | 04/13 07:17 |
| $E_6$ | U.S. supports Japan's Fukushima wastewater disposal decision | 199 | 04/13 10:19 |
| $E_7$ | Swedish green girl responds vaguely to Japanese incident | 56 | 04/13 18:30 |
| $E_8$ | Japanese government makes radioactive tritium mascot | 143 | 04/13 21:22 |
| $E_9$ | U.S. bans Japanese food from entering | 171 | 04/13 23:05 |
| $E_{10}$ | Japan's deputy prime minister says it's OK to treat nuclear wastewater | 198 | 04/14 07:59 |
| $E_{11}$ | Japan announces removal of radioactive tritium mascot | 101 | 04/15 05:29 |
| $E_{12}$ | Zhao Lijian asks Japanese politicians to cook and wash with nuclear wastewater | 75 | 04/15 16:00 |
| $E_{13}$ | Korean people black out Japanese seafood | 61 | 04/15 18:06 |
| $E_{14}$ | Japan bans on sale of Fukushima black scorpion fish | 98 | 04/20 06:48 |
| $E_{15}$ | Japan wants to put Fukushima ingredients on the Olympic table | 49 | 04/21 14:30 |

messages from $m^i$ and extract all nouns in them. The nouns set is denoted as $\mathcal{N} = \{N_1, N_2, \ldots\}$. We then build an inverted index where every noun in $\mathcal{N}$ is mapped to $E_i$ and iterate through this step to update the inverted index until all events are computed. Next, we retrieve the events which are possibly related to $E_i$ and occur before $E_i$ from the inverted index, which may filter out the most irrelevant events. Algorithm 1 illustrates the method to generate event evolution graph $\mathcal{G}$ for given events set $\mathcal{E}$.

## IV. EXPERIMENT

In this section, we introduce the details of the experiments and analyse the results.

### A. Dataset and Experimental Settings

We collected Weibo messages of which post time is from Mar. 1st to Apr. 30th in 2021. There are about 10,000 events detected through our event detection algorithm.

For data processing, we apply LTP [27] for Chinese word segmentation and TexSmart [28] for POS tagging/NER. We extract keywords and summaries from events through TextRank algorithm [29]. The word embeddings used to measure the similarity of keywords come from "Chinese Word Vectors" [30], which are trained by SGNS on Weibo data, and the dimension of word embedding is $d = 300$. The word embeddings used to measure the similarity of summary and posts are generated by the incremental trained model based on "chinese-roberta-wwm-ext" [31]. We use about 1 million Weibo data for unsupervised incremental training.

To demonstrate our method better, we choose a topic of events in this paper, i.e., "Japan's nuclear wastewater discharge into the sea." There are 1,731 posts and 15 events under this topic. The period of events is from Apr. 9th to Apr. 21st. Table II shows the details of these events.

### B. Ground Truth

We invite three annotators to read the selected events in Table II and the social messages they contained. Then the annotators construct the event evolution graph independently. We consider the three graphs comprehensively, and the final graph can be regarded as event evolution relationships close to the objective existence. As shown in Fig. 2, the ground truth contains 22 edges, which reveal the real evolution relationship between events.

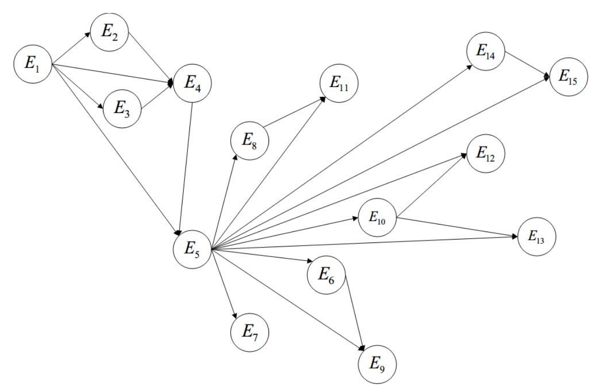


Fig. 2. The Ground Truth Event Evolution Graph

### C. Evaluation Metrics

Assuming $\mathcal{G} = \{\mathcal{E}, \mathcal{L}\}$ is the event evolution graph created artificially, which is regarded as ground truth. $\mathcal{G}' = \{\mathcal{E}, \mathcal{L}'\}$ is generated by our method automatically. $\mathcal{E}$ is the set of nodes in the graph, i.e., the set of events under the given topic in this paper. $\mathcal{L}$ and $\mathcal{L}'$ are the sets of edges in the graph, i.e., the sets of event evolution relationships. Our purpose is to minimize the difference between $\mathcal{L}$ and $\mathcal{L}'$.

Since the process of constructing event evolution graph can be regarded as a binary classification problem for each edge, we adopt the measurement of **Precision**($P$), **Recall**($R$) and **F1-score**($F1$) for our evaluation which are used in [32].

**Precision**($P$) is the ratio of the number of correct evolution relationships in the graph constructed automatically to the number of total evolution relationships constructed with our method, as is formalized in (11).

$$P = \frac{|\mathcal{L} \cap \mathcal{L}'|}{|\mathcal{L}'|} \tag{11}$$

**Recall**($R$) is the ratio of the number of correct evolution relationships in the graph constructed automatically to the number of total evolution relationships in the graph constructed artificially, as is formalized in (12).

$$R = \frac{|\mathcal{L} \cap \mathcal{L}'|}{|\mathcal{L}|} \tag{12}$$

**F1-score**($F1$) is a metric which takes both precision and recall into account, as is formalized in (13).

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{13}$$

### *D. Results and Analysis*

**Baseline.** The method of generating event evolution graph based on similarity measurement is used in most previous works. The difference between them lies in the factors considered. We adopt the method proposed in [33] as our baseline, which considers the similarity of the posts, location and participants related to events. The event similarity score function in the baseline method is defined as follows:

$$\begin{aligned} Sim(E_i, E_j) = \beta \cdot Sim_{loc}(loc^i, loc^j) + \\ \gamma \cdot Sim_{par}(par^i, par^j) + \omega \cdot Sim_m(m^i, m^j) \end{aligned} \tag{14}$$

where $\beta, \gamma, \omega$ are weight coefficients and can be adjusted with the requirement of $\beta + \gamma + \omega = 1$. We set the threshold as $\delta = 0.5$ in this paper and $\delta = 0.4$ in the baseline method.

TABLE III
RESULT OF DIFFERENT WEIGHTS

| | $\alpha$ | $\epsilon$ | $\rho$ | $\beta$ | $\gamma$ | $\omega$ | P | R | F1 |
|---|---|---|---|---|---|---|---|---|---|
| Our Method | 0 | 6 | 3 | 1 | 0 | 0 | 0.533 | 0.727 | 0.615 |
| | 1 | 6 | 3 | 0 | 0 | 0 | 0.455 | **0.909** | 0.606 |
| | 1 | 6 | 2 | 1 | 0 | 0 | 0.516 | 0.727 | 0.604 |
| | 1 | 5 | 1 | 1 | 1 | 1 | 0.667 | 0.727 | 0.696 |
| | 1 | 5 | 2 | 1 | 1 | 0 | 0.696 | 0.727 | **0.711** |
| | 1 | 4 | 2 | 1 | 2 | 0 | **0.800** | 0.546 | 0.649 |
| | 1 | 1 | 1 | 1 | 4 | 2 | 0.609 | 0.636 | 0.622 |
| | 1 | 1 | 2 | 1 | 4 | 1 | 0.583 | 0.636 | 0.609 |
| | 1 | 4 | 1 | 2 | 1 | 1 | 0.667 | 0.727 | 0.696 |
| | 1 | 4 | 2 | 2 | 1 | 0 | 0.625 | 0.682 | 0.652 |
| | 2 | 4 | 0 | 1 | 2 | 1 | 0.722 | 0.591 | 0.650 |
| | 2 | 4 | 1 | 1 | 2 | 0 | 0.778 | 0.636 | 0.700 |
| Baseline Method | | | | 1 | 2 | 7 | 0.275 | 0.636 | 0.384 |

We manually traverse all groups of weights $\alpha, \epsilon, \rho, \beta, \gamma, \omega$ by $step = 0.1$, and choose some representative combinations to show in Table III. Each weight is multiplied by 10 in order to display correctly in the table. F1-score is the highest when $\alpha = 1, \beta = 1, \gamma = 1, \epsilon = 5, \rho = 2, \omega = 0$, which means $Sim_{key}$ plays the most important role in event similarity measurement, and $Sim_m$ plays the least role. In most weights settings, our method outperforms the baseline in three indicators, which means taking semantics into account really works. Unlike the baseline method, we use keywords and summaries of events for calculating semantic similarity. The two parts pay more attention to the relevance and difference of two events, and give up some useless imformation in the posts. Meanwhile, we notice that the value of $\alpha$ is not suitable for setting too large, mainly because time is a pretty uncertain factor. In today's public opinion environment, events can cause much discussion in a very short time, which leads to other events. On the other hand, two events which are separated for a long time may be related for some reason.

## V. CONCLUSION

This paper proposes a novel method to analyze the evolution of events detected from social messages. Concretely, we use a 6-tuple to represent events, and based on this, we define an event similarity measure function with adjustable parameters. In order to better calculate the content similarity, we apply pre-trained word embedding to introduce semantic information. According to the event similarity measure function, we generate event evolution graphs on a Weibo dataset. Our experiment shows that our method performs better than the baseline method.

In future work, we plan to introduce more information related to events by combining knowledge graph. In addition, reinforcement learning (RL) may be effective for parameter optimization.

## ACKNOWLEDGMENT

The authors of this paper were supported by the National Key Research and Development Program of China under the Grant No.2018YFC0830804, and S&T Program of Hebei through the Grant No.21340301D.

## REFERENCES

[1] H. Peng, J. Li, Y. Song, R. Yang, R. Ranjan, P. S. Yu, and L. He, "Streaming social event detection and evolution discovery in heterogeneous information networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–33, 2021.

[2] Y. Cao, H. Peng, J. Wu, Y. Dou, J. Li, and P. S. Yu, "Knowledge-preserving incremental social event detection via heterogeneous gnns," *arXiv preprint arXiv:2101.08747*, 2021.

[3] W. Yu, J. Li, M. Z. A. Bhuiyan, R. Zhang, and J. Huai, "Ring: Real-time emerging anomaly monitoring system over text streams," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 506–519, 2017.

[4] Y. Liu, H. Peng, J. Guo, T. He, X. Li, Y. Song, J. Li *et al.*, "Event detection and evolution based on knowledge base," in *Proc. KBCOM*, 2018.

[5] Q. Mao, X. Li, H. Peng, J. Li, D. He, S. Guo, M. He, and L. Wang, "Event prediction based on evolutionary event ontology knowledge," *Future Generation Computer Systems*, vol. 115, pp. 76–89, 2021.

[6] Y. Gao, L. Xiaoyong, P. Hao, B. Fang, and P. Yu, "Hincti: A cyber threat intelligence modeling and identification system based on heterogeneous information network," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[7] Y. Liu, H. Peng, J. Li, Y. Song, and X. Li, "Event detection and evolution in multi-lingual social streams,"

*Frontiers of Computer Science*, vol. 14, no. 5, pp. 1–15, 2020.

[8] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, “Large-scale hierarchical text classification with recursively regularized deep graph-cnn,” in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1063–1072.

[9] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, P. Yu, and L. He, “Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification,” *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[10] Y. He, J. Li, Y. Song, M. He, H. Peng *et al.*, “Time-evolving text classification with deep neural networks.” in *IJCAI*, 2018, pp. 2241–2247.

[11] Y. He, Y. Song, J. Li, C. Ji, J. Peng, and H. Peng, “Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 639–648.

[12] Y. Cao, H. Peng, and S. Y. Philip, “Multi-information source hin for medical concept embedding,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2020, pp. 396–408.

[13] H. Peng, J. Li, Z. Wang, R. Yang, M. Liu, M. Zhang, P. S. Yu, and L. He, “Lifelong property price prediction: A case study for the toronto real estate market,” *arXiv preprint arXiv:2008.05880*, 2020.

[14] S. Zhu, J. Li, H. Peng, S. Wang, P. S. Yu, and L. He, “Adversarial directed graph embedding,” *arXiv preprint arXiv:2008.03667*, 2020.

[15] H. Peng, J. Li, H. Yan, Q. Gong, S. Wang, L. Liu, L. Wang, and X. Ren, “Dynamic network embedding via incremental skip-gram with negative sampling,” *Science China Information Sciences*, vol. 63, no. 10, pp. 1–19, 2020.

[16] G. Luo, J. Li, H. Peng, C. Yang, L. Sun, P. S. Yu, and L. He, “Graph entropy guided node embedding dimension selection for graph neural networks,” *arXiv preprint arXiv:2105.03178*, 2021.

[17] H. Peng, J. Li, Q. Gong, Y. Ning, S. Wang, and L. He, “Motif-matching based subgraph-level attentional convolutional network for graph classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5387–5394.

[18] Z. Deng, H. Peng, D. He, J. Li, and P. S. Yu, “Htcinfomax: A global model for hierarchical text classification via information maximization,” *arXiv preprint arXiv:2104.05220*, 2021.

[19] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang *et al.*, “Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting,” *Information Sciences*, vol. 521, pp. 277–290, 2020.

[20] J. Li, H. Peng, Y. Cao, Y. Dou, H. Zhang, P. Yu, and L. He, “Higher-order attribute-enhancing heterogeneous graph neural networks,” *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[21] B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li, “Deep irregular convolutional residual lstm for urban traffic passenger flows prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2019.

[22] Z. Zhene, P. Hao, L. Lin, X. Guixi, B. Du, M. Z. A. Bhuiyan, Y. Long, and D. Li, “Deep convolutional mesh rnn for urban traffic passenger flows prediction,” in *2018 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*. IEEE, 2018, pp. 1305–1310.

[23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *arXiv preprint arXiv:1802.05365*, 2018.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[26] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, “Fine-grained event categorization with heterogeneous graph convolutional networks,” *arXiv preprint arXiv:1906.04580*, 2019.

[27] W. Che, Y. Feng, L. Qin, and T. Liu, “N-ltp: A open-source neural chinese language technology platform with pretrained models,” *arXiv preprint arXiv:2009.11616*, 2020.

[28] H. Zhang, L. Liu, H. Jiang, Y. Li, E. Zhao, K. Xu, L. Song, S. Zheng, B. Zhou, J. Zhu *et al.*, “Texsmart: A text understanding system for fine-grained ner and enhanced semantic analysis,” *arXiv preprint arXiv:2012.15639*, 2020.

[29] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[30] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, “Analogical reasoning on chinese morphological and semantic relations,” *arXiv preprint arXiv:1805.06504*, 2018.

[31] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, “Revisiting pre-trained models for chinese natural language processing,” *arXiv preprint arXiv:2004.13922*, 2020.

[32] C. C. Yang, X. Shi, and C.-P. Wei, “Discovering event evolution graphs from news corpora,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 4, pp. 850–863, 2009.

[33] Z. Lu, W. Yu, R. Zhang, J. Li, and H. Wei, “Discovering event evolution chain in microblog,” in *2015 IEEE HPCC*. IEEE, 2015, pp. 635–640.