Comprehensive Event Storyline Generation from Microblogs

Wenjin Sun
17120411@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China
CETC Big Data Research Institute Co.,Ltd.
Guiyang, China

Yuhang Wang
19120421@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Yuqi Gao
gaoyq@smail.nju.edu.cn
Nanjing University
Nanjing, China

Zesong Li
lizesong@cetcbigdata.com
CETC Big Data Research Institute Co.,Ltd.
Guiyang, China

Jitao Sang
jtsang@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

Jian Yu
jianyu@bjtu.edu.cn
Beijing Jiaotong University
Beijing, China

ABSTRACT

Microblogging data contains a wealth of information of trending events and has gained increased attention among users, organizations, and research scholars for social media mining in different disciplines. Event storyline generation is one typical task of social media mining, whose goal is to extract the development stages with associated description of events. Existing storyline generation methods either generate storyline with less integrity or fail to guarantee the coherence between the discovered stages. Secondly, there are no scientific method to evaluate the quality of the storyline. In this paper, we propose a comprehensive storyline generation framework to address the above disadvantages. Given Microblogging data related to the specified event, we first propose Hot-Word-Based stage detection algorithm to identify the potential stages of event, which can effectively avoid ignoring important stages and preventing inconsistent sequence between stages. Community detection algorithm is applied then to select representative data for each stage. Finally, we conduct graph optimization algorithm to generate the logically coherent storylines of the event. We also introduce a new evaluation metric, SLEU, to emphasize the importance of the integrity and coherence of the generated storyline. Extensive experiments on real-world Chinese microblogging data demonstrate the effectiveness of the proposed methods in each module and the overall framework.

CCS CONCEPTS

• **Information systems → Data mining**.

MMAsia '19, December 15–18, 2019, Beijing, China

ACM ISBN 978-1-4503-6841-4/19/12.
https://doi.org/10.1145/3338533.3366601

KEYWORDS

Social media, event detecting, microblog, storyline generation, graph optimization, community detection

ACM Reference Format:
Wenjin Sun, Yuhang Wang, Yuqi Gao, Zesong Li, Jitao Sang, and Jian Yu. 2019. Comprehensive Event Storyline Generation from Microblogs. In *ACM Multimedia Asia (MMAsia '19), December 15–18, 2019, Beijing, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3338533.3366601

1 INTRODUCTION

The ever-growing number of people using social media service makes it a valuable source of timely information. For example, Microblogging data contains a wealth of information of trending events, e.g. media reports, user comments and discussions, etc. Event storyline generation is one typical task of social media mining, whose goal is to extract the development stages with associated description of events. Specifically, when a user finds a trending event on the page of Trending (where events are described with only one sentence or several words), he can only access to the event information at that stage. Microblog may be flooded with information of the current stage at the time. Faced with the chaotic data, it is difficult for general users to get information of other stages and obtain the overall development of the event manually. People would get lost in local information with high probability, and cannot grasp the global information of trending events quickly. However, people can browse the detailed development of the event conveniently without facing the messy data on the network with storyline generation method.

In this paper, trending events refer to events that last longer, develop more twists and turns, and attract a lot of people. The task of event storyline generation from microblogs is to extract a understandable storyline of the trending event from event-related microblogging data.

Several researchers have focused on related problems. Methods for storyline generation are proposed for short[2][11][13]or long text [8][3][12][7]. For microblog, a typical short text data, the existing models can be mainly divided into two categories, clustering-based and graph-based optimization approaches. Stages of event are too difficult to distinguish by general cluster-based algorithm due to

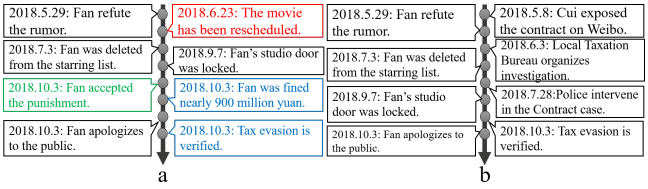

Figure 1: Samples storyline for "Bingbing Fan's Incident"

the high similarity between consequent stages. Even with effective clustering, it is impossible to connect the stages into a smooth storyline. Graph-based optimization algorithms can generate relatively coherent storylines, but the diversity of data cannot be guaranteed. Taking "Bingbing Fan's Incident" as an example, as shown in Figure 1(a), it cannot detect the stage of "Cui expose contract", which is an important stage of the event. Then some stages, marked by the blue box, may be detected multiple times. The stage of "Movie has been rescheduled" marked by red box in (a) is event-unrelated content, which would make people more confused about events. Some stages disorder, marked by green box in (a). In this paper, we address the limitations in current storyline generation methods and propose a framework for generating comprehensive event storylines satisfying the following two requirements: (1)Integrity, covering every stage of development of the event; (2)Coherence, stages extracted are arranged in a certain order, with a development between the stages and there is no event-unrelated content in the storyline. The derived comprehensive storyline is illustrated in Figure1(b).

We design a Comprehensive Event Storyline Generation framework in this paper to analyze the development of trending events. As show in Figure2, our framework consists of three stages: (1) Hot-Word-Based stage detection algorithm is proposed to clarify the possible stages of event, which can effectively avoid ignoring important stages. (2) We filter out noise data with the information of keywords and apply Community Detection algorithm to select representative data, preventing stage redundancy and inconsistent sequence between stages. (3) We conduct graph optimization algorithm to generate the storylines of the event, which can generate logically coherent storylines. Besides, there are no specialized quantitative methods of storyline quality evaluation. The existing methods for storyline generation is evaluation toolkits for document summarization. They treat the storyline as a summary to evaluate the quality of a storyline on word level, ignoring the temporal and structural information, i.e. integraity and coherence of stages, which is important for storyline. We propose a new metric method, named SLEU, to evaluate the quality of a storyline on sentence (stage) level, which can comprehensively evaluate the integrity and coherence of the storyline. Thus, the contributions of this article are as follows:

1) We design a Comprehensive Event Storyline Generation framework to extract a storyline from event-related microblogging data with the feature of integrity and content coherence.
2) A more scientific method, SLEU, is proposed to evaluate the storyline.
3) Extensive experiments on real-world Chinese microblogging data demonstrate the effectiveness of the proposed methods in each module and the overall framework.

The rest of the paper is organized as follows. The related work on event storyline generation is presented in Section2. Section3 elaborates our methodology for comprehensive event storyline generation. In Section 4, we present evaluation and experiment. Finally, Section 5 concludes the paper.

2 RELATED WORK

2.1 Clustering-based Approaches

Story Forest (i.e. a set of online schemes that automatically clusters streaming documents into events) is proposed in [8]. They connect related events in growing trees to tell evolving stories. Two-layer document clustering procedure is applied to generate story trees. However, they utilize a semi-supervised document clustering procedure to determine whether the two documents belong to the same event, which would be impossible for short text. Temporal and textual clustering approach is applied in [2] to obtain multifaceted clues about the event, which is also unsuitable for microblog because the data associated with the same event is too similar to separate. In [12], a similarity calculation method is proposed to retrieve news articles related to the specific event, which combines textual similarity, temporal similarity and entity similarity. An online tweet stream clustering algorithm and topic evolvement detection method is applied in [11] to produce timelines automatically from tweet streams. The similarity of the data between each stage is too high, so it is too difficult to distinguish by general cluster-based algorithm. Even with effective clustering, it is impossible to connect the stages into a content coherent storyline.

2.2 Graph-based Optimization Approaches

Storyline generation from microblog via graph optimization is first applied in [4]. They apply an IR model to retrieve relevant tweets of an event. But it is obviously unreasonable to use the event query, which is a set of user defined keywords or phrases, to detect the Burst Time and calculate the weight of the points in the multi-view graph. Because it will lead to a reduction of the diversity of the data during the optimization process, and eventually ignore many important stages of the event. The key idea of [13] is the utilization of an approximate solution for the dominating set problem. In a word, it is impossible to detect all stages of the event and arrange them in a certain order via graph optimization due to the sparse, noisy nature of microblogs.

3 FRAMEWORK DESCRIPTION

In this section, we describe our framework detailed. Figure2 presents an overview of the framework. The input of our framework is microblogging data about an event. Table 1 delivers the definitions of the notations used in this section. Below we will give a detailed introduction to each module of the framework.

3.1 Stage Detection

For trending events, there will be a large number of discussions in Microblog. As the event evolves, the number of people participating in the discussion and the key points discussed by the people will change accordingly due to the immediacy of microblog. Such features can be used for storyline generation. It is difficult to detect

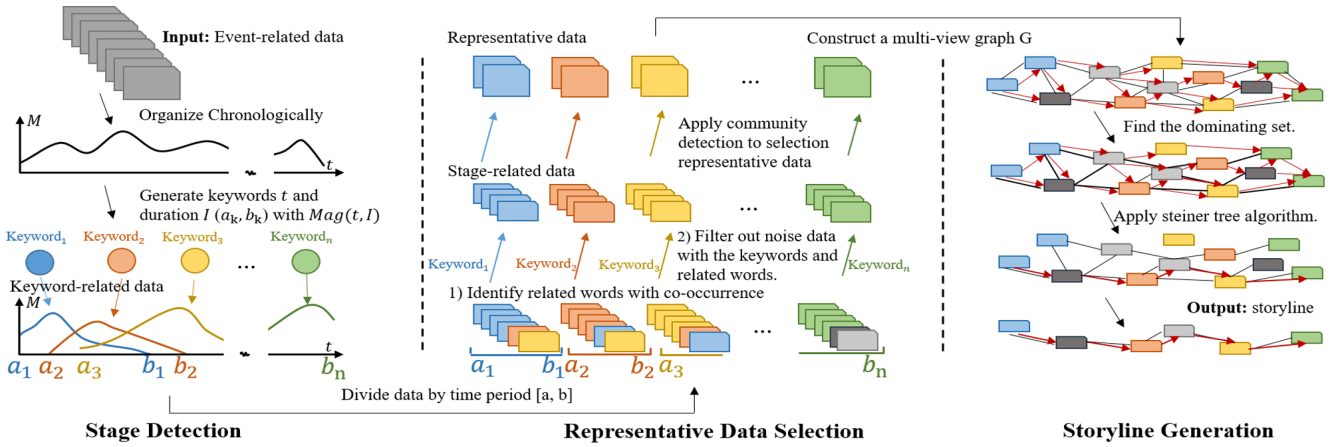

Figure 2: The overview of the framework

Table 1: Table of notations

Notation	Definition
M	Total number of data in the corpus
M^i	Number of data in the i^{th} time-slice
M_t^i	Number of data in the i^{th}time-slice that contain the word t
m	Number of representative data of the event
m_j	Number of representative data of the j^{th} stage
M_j	Number of data in the j^{th} stage
M^*	Number of data after filtering
M_j^*	Number of filtered data in the j^{th} stage

every stage of events only by focusing on people's attention (i.e. the amount of data). We also need combine the unique features of each stage. Some keywords will be more frequent at a particular stage of development than at other stages, we assumed that the frequency of the word during this time period is anomalous and can be utilized as a feature to distinguish this stage, namely the keywords of this stage. We expect that the keywords could reflect the feature of the stage best and be different from the keywords from other stages. Therefore, we propose Hot-Word-Based stage detection algorithm to clarify the stages of event. We determine the keywords and duration of each stage referring to [1].

Our purpose of this subsection is to identify duration and keywords of stage. The input of our framework is the microblog data collection related to the trending event. We first use NLP tool to perform word segmentation on the textual data. The output of this subsection is a list L, where $|L| = n$, containing the most important n stages of the event. The description of each stage consists of keywords and its time periods.

3.1.1 Word anomaly measurement. We calculate the difference of the word occurrence frequency between inside the entire data set and inside a certain time slice referring to [1]. The author assumes that the number of tweets that contain the word t in the i^{th} time-slice, M_t^i, follows a generative probabilistic model. For a corpus large enough, it seems reasonable to model this kind of probability with a binomial distribution, as shown in Formula (1).

$$P(M_t^i) = \begin{pmatrix} M^i \\ M_t^i \end{pmatrix} p_t^{M_t^i} (1-p_t)^{M^i - M_t^i} \tag{1}$$

Because M^i is large, they further assume that $P(M_t^i)$ can be approximated by a normal distribution, i.e. $P(M_t^i) \sim N(M^i p_t, M^i p_t(1-p_t))$. Then we can calculate the expected frequency of tweets containing the word t in the i^{th} time-slice, $E(t/i) = M^i p_t, where\ p_t = M_t/M$. Finally, the difference between the two values is used to express the abnormality of word t in time slice i, $anomaly(t,i) = M_t^i - E[t/i]$.

3.1.2 Keyword extraction. We calculate the $Mag(t, I)$ to determine the keywords t and duration I of each stage. It corresponds to the algebraic area of the anomaly function on $[a, b]$ (a, b indicates the beginning and end of stage respectively), as shown in Formula(2).

$$Mag(t, I) = \int_a^b anomaly(t, i)\, di = \sum_{i=a}^{b} anomaly(t, i) \tag{2}$$

The algebraic area is obtained by integrating the discrete anomaly function, which in this case boils down to a sum.

3.1.3 Stage identification. We maximize $Mag(t, I)$ for each word t by traversing every time interval and then determine the keywords and duration of each stage as shown in Formula (3)(4).

$$I = \underset{I}{argmax}\, Mag(t, I) \tag{3}$$

$$Mag(t, I) = max(\sum_{i=a}^{b} anomaly(t, i)\,(1 \le a \le b \le s)) \tag{4}$$

Finally, we merge duplicated stages whose time interval contain too many coincident fragments. So, we apply the parameter θ to measure the coincidence of the two stages. When the quotient of the intersection and union of the two time intervals are greater than θ, we merge the two stages. The keywords of merged stages

are the keywords of both of them, and time interval is the union of the two time intervals.

3.2 Representative Data Selection

From the previous subsection we have obtained the duration $[a, b]$ and corresponding keyword t of each stage, then we need to select representative data of each stage for storyline generation. In addition to reducing the complexity of the algorithm, we need filter out the noise data in this subsection. For some stages with fewer discussions, the noise data may be more than effective data. If we don't select representative data, the accuracy of storylines can not be guaranteed. We should choose the data that 1) best characterizes this stage, that is, the content is representative and relevant to the keywords. 2) For the stage with more data, we hope to select more representative data. 3) There should be some differences between these representative data of every stage, so that the generated storyline will be more diverse and readable.

Therefore, we first determine related words, filter the data of each stage according to the keywords and related words, and remove the data unrelated to the stage to extract the data that best represents the stage. Then we create a graph, and cluster the data of each stage by community detection algorithm. We select the center of each community as representative data.

3.2.1 Related words generation and information filtering. We apply word co-occurrence to determine the relationship between words. The more co-occurring with the keyword, the closer the relationship with the keyword is, the more likely the word is to represent the relevant words at this stage. Therefore, we refer to [1] to use $\rho o_{t,t'}$ to measure the correlation of the words t and t', as shown in (5).

$$\rho o_{t,t'} = \frac{\sum_{i=a+1}^{b} A_{t,t'}}{(b-a-1)A_t A_{t'}} \tag{5}$$

$$A_{t'}{}^2 = \frac{\sum_{i=a+1}^{b} (M_{t'}^i - M_{t'}^{i-1})^2}{b-a-1}, \; A_{t,t'} = (M_t^i - M_t^{i-1})(M_{t'}^i - M_{t'}^{i-1})$$

The changes of related words will be consistent with the changes of the keywords. A larger value indicates a closer correlation. We use this method to find top R words related to the keywords as related words. The cosine similarity is used to calculate the distance between the $R+1$ words and each piece of data, and the data whose distance is less than a threshold β is removed.

3.2.2 Representative data extraction. We construct a graph whose nodes represent the data and join the two nodes by an edge if and only if the text similarity between the two responding data is greater than distance α_1, which is calculated with cosine similarity. Then we introduce parameter m, m_j, which are described in Table 1. m_j is determined according to the amount of data in each stage, $m_j = \frac{M_j^*}{M^*} * m$. We utilize the betweenness centrality score[10] of edges to measure the strength of each edge in the graph. An edge's betweenness score is defined as the number of shortest paths between all pairs of nodes that pass through it. An edge between two communities is expected to achieve a high betweenness score. Edges with high betweenness score will be removed iteratively to extract communities. The iterative splitting process will stop until the number sub-graph is greater than or equal to m_j. Then, the required m_j data and the center of each community are extracted and selected as representative data.

3.3 Storyline Generation

The representative data of each stage of the event have been extracted above, and a coherent storyline will be generated using graph optimization referring to [4]. A multi-view graph $G = (V, W, E, A)$ is constructed, where V is a set of vertices (nodes), indicating representative data. E is a set of undirected edges, which represents the similarities between pieces of data, we join the two nodes by an edge if and only if the text similarity between the two responding data is greater than α_2, which is calculated with cosine similarity. W is the weights of V, which is used to measure the representativeness and uniqueness of each piece of data. A is a set of directed edges, which represents the time continuity of the data. we draw an arc from v_i to v_j if and only if $\tau_1 \le t_j - t_i \le \tau_2$, where t_i and t_j are their time stamps respectively. We call $[\tau_1, \tau_2]$ the temporal window.

We first find the dominating set on the undirected graph $G = (V, W, E)$ (i.e., without considering A in the multi-view graph), and then perform the steiner tree algorithm to connect the dominating set on the directed graph $G = (V, W, A)$ (i.e., without considering E in the multi-view graph) which takes the time continuity into consideration and leads to a coherent storyline. [1]

The difference between this paper and [4] is the weight of the notes in the graph. In [4], the weight of the node is calculated with the similarity of each point to the keywords. This reduce the diversity of the data significantly, leading to incomplete storyline. Since our purpose is to further filter the data and avoid data redundancy as much as possible, the weight of the point is to measure the representativeness of each piece of data. The node with more neighbors or more similar with the neighbor should be given a small weight. Therefore, we define the weight of each point to be the reciprocal of the number of the its neighbors. In this way, a stage that has not been detected in the previous two steps would be detected with high probability, that is, even if the degree of relevance to the keyword is not high, it is still possible to be left for storyline generation.

4 EXPERIMENT

In this section, we evaluate the performance of the proposed framework. In particular, Section 4.1 introduces the evaluation metric of the experiment. Dataset and baselines for comparison is described in Section 4.2. Section 4.3 and 4.4 presents the results of experiment, including qualitative and qualitative evaluation of our framework in each module and the overall framework.

4.1 Evaluation metric

4.1.1 ROUGE. Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[5] is an evaluation toolkit for document summarization, which is widely applied to evaluate the quality of storyline generation, such as [4][11]. Among supported metrics, ROUGE-1 has been demonstrated to be the most consistent with human judgement[6]. Comprehensive comparison of the evaluation methods of other

[1] Due to limited space, here is a brief introduction, please refer to [4] for details.

models of storyline generation, we validate that ROUGE-1 is suitable for measuring storyline extracted from microblog.

4.1.2 SLEU. Most of researchers apply the evaluation metrics of document summarization to measure the quality of storylines, such as ROUGE. The storyline are treated as a summary to evaluate its quality on word level, ignoring the temporal and structural information, i.e. integraity and coherence of stages, which is important for storyline. We propose a novel metric named SLEU (StoryLine Evaluation Understudy) based on BLEU (Bilingual Evaluation Understudy), which is proposed for machine translation [9]. The sentences in the storyline which is generated from the system are s_1, s_2, ... s_n, where s_1 is the beginning of the storyline and s_n is the end. Then, sentences of reference storylines are s'_1, s'_2, ... s'_n. n is the number of stages. The *h-gram* and *h-gram′* here represent h sentences sequences in a storyline, which is not necessarily continuous, generated from reference storyline and generated storyline respectively. We use the collocation information between sentences in the storyline to calculate the probability of the storyline, so as to judge whether a storyline is complete or coherent. We first calculate the cosine similarity between $s_i (i \in [1, n])$ and $s_j (j \in [1, n])$. Then we compute the probability, p_h, of *h-gram* matches between the two storylines separately, as shown in (6).

$$p_h = \frac{\min(Count_{match}(h-gram), Count_{match}(h-gram'))}{Count(h-gram)} \quad (6)$$

The values that h can take depends on the situation, and is represented by the set H. A storyline using the 1-*grams* tends to satisfy comprehensive. The longer *h-gram* matches account for coherence. $Count(h\text{-}gram)$ indicates the number of comparable *h-gram* (e.g. if h=4, $Count(2\text{-}gram) = 6$). $Count(h\text{-}gram')$ is equal to $Count(h\text{-}gram)$. $Count_{match}(h\text{-}gram)$ indicates the number of *h-gram* that can be matched (i.e. the similarity of the corresponding sentences in the *h-gram* pair is more than a specific threshold λ, in our experiment, we set λ=0.2). Then we can calculate SLEU = $\exp(\sum_{h \in H} w_h \log p_h)$. The SLEU metric ranges from 0 to 1. In this way, we can quantitatively evaluate the storyline. Experiments have shown that SLEU is highly correlated with human evaluation.

4.1.3 User study. Since storyline generation is a subjective process, to better evaluate the generated storylines, we conduct a user survey. We enlisted 20 human reviewers who are senior undergraduate students that not related to this experiment, to blindly evaluate the results given by different approaches. Each individual storyline was reviewed by 20 different reviewers. They mainly evaluate from the following aspects: 1) Whether the information coverage is comprehensive (i.e.whether important content has been ignored or not)? 2) Whether the content is logically coherent? Is it consistent with the development of the event? Are there repeating stages? Whether there are stages that are not related to the event. 3) Whether the overall storyline is easy to read and understand? They score each aspect separately and finally calculate the average score to evaluate the quality of the storyline.

4.2 Dataset and Baseline

4.2.1 Dataset. Since the Twitter or other microblogs corpora used in prior work aren't available, we evaluated our methods by running them on real data. We crawled about 10 million pieces of data from Sina Weibo, including posts and comments. Then we apply the combination of retrieval to extract event-related data, covering 10 hot events which were selected in the hot search list including Bingbing's Fan tax evasion, Chongqing bus crash, etc. The ground truth of storylines is extracted from dsj365.cn, which is a news website, whose content is edited by journalists.

4.2.2 Baselines for Comparison. We consider three methods for comparison: **Sumblr**[11], a method for continuous summarization of evolving tweet streams, produce timelines automatically from tweet streams and an online stream clustering algorithm is applied to generate storyline. Since the task in this paper has no user query, we delete pyramidal time frame of Sumblr for better comparison with us. [2], named **TTC** in the rest of paper, applies the temporal and textual clustering approach to obtain multifaceted clues about the event and to characterize the intra-evolution of each clue. Because we extracted data from the time and content of the post, and did not involve the relationship data of Microblog users, we delete inter-clue connection of the frame. Cluster-based methods are applied by [11] and [2] to generate storylines. [4], named **QGO** in the rest of paper, applies a retrieve-based approach to filter data before applying graph-based optimization to generate storylines. We apply topic model to generate query instead of user query in the model for better comparison with us.

To further evaluate each mudule of the framework, we design four comparative experiments: 1) The stage detection algorithm in the framework is replaced by the temporal and textual clustering approach to verify the effectiveness of the stage detection algorithm, named **CESG-S**; 2) We remove the community detection algorithm in step 2 and leave the other parts unchanged to demonstrate the effectiveness of it on data diversity, named **CESG-CD**; 3) Replace the community detection algorithm in step 2 with page-ranking algorithm to select representative data, and the other parts remain unchanged, to illustrate the effectiveness of community detection algorithm on data diversity, named **CESG-PR**; 3) In order to show the necessity of data filtering i.e. the first two steps of the step 2, we remove them and remain the other part unchanged, named **CESG-F**. Our full framework is named **CESG** in the rest of paper.

4.3 Qualitative Experiment

4.3.1 Comparative experiments. As described above, we design four comparative experiments. Due to the limited space, we take "Jinfu's violence incident" as an example to analyze the results. We cut the generated storyline and translate key information into English for clear presentation, as shown in Figure3.

Stages marked with red are associated with events development, and the sequence are right, i.e. that is effect information. The sequence of stages marked by green are wrong. Stages with no mark are repetitive or event-unralated. From the experimental results, we can intuitively see the number of red stages in CESG is more than others. Some important stages would be ignored, or the sequence disorder as shown in CESG-S, CESG-CD, CESG-F and CESG-PR. For example, the stage of "Jiang's friend sent an article again to support

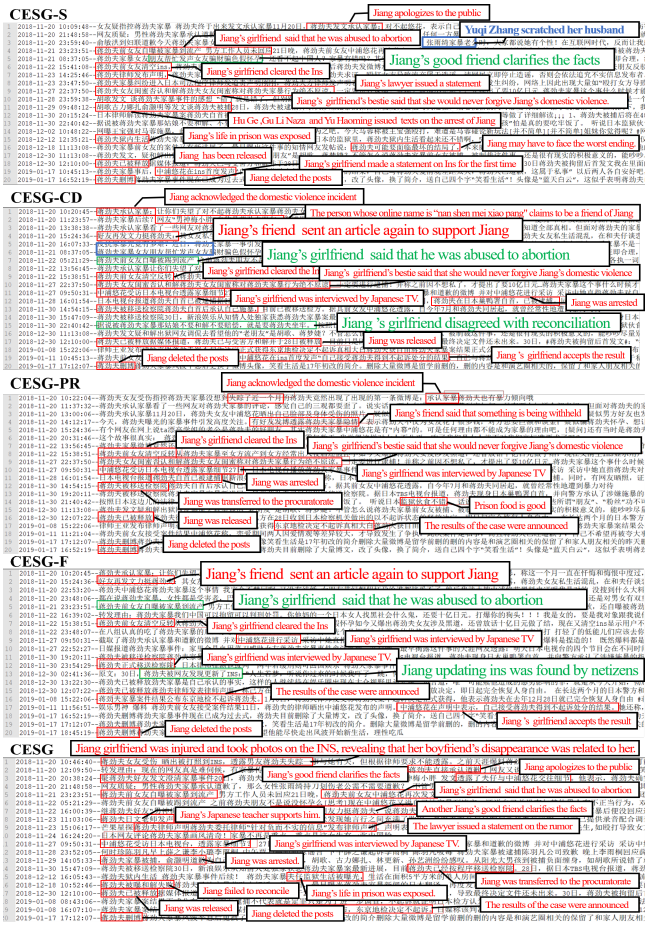

Figure 3: Storylines generated from the comparative models.

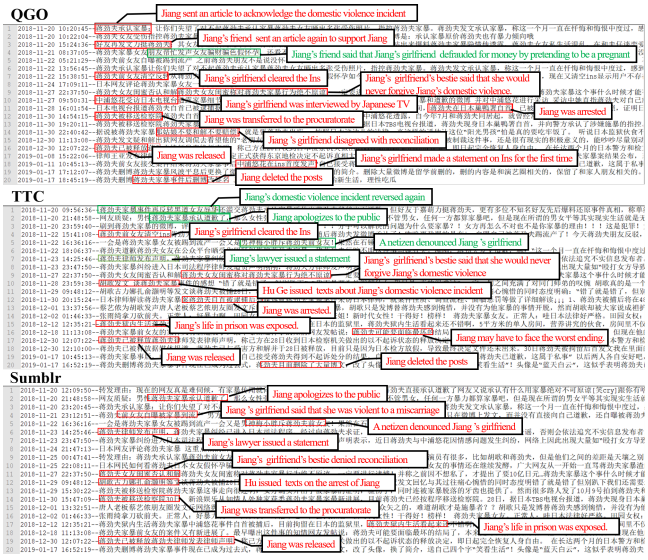

Figure 4: Storylines generated from the other three models.

Table 2: The result of all the experiments

	User score	SLEU	GOUGE
QGO	0.533	0.128	0.139
TTC	0.411	0.108	0.184
Sumblr	0.529	0.148	0.203
CESG-S	0.394	0.137	0.176
CESG-CD	0.522	0.204	0.201
CESG-PR	0.524	0.219	0.195
CESG-F	0.489	0.189	0.179
CESG	0.827	0.285	0.307

Jiang" cannot be detect in CESG-S. All of the four methods cannot detect the stage of "Jiang girlfriend was injured and took photos on the INS", which is the first stage of the event. The sequence of stages, "Jiang's girlfriend said that he was abused to abortion" and "Jiang's friend sent an article again to support Jiang" is wrong in CESG-CD. Some contents unrelated to the event may be extracted, such as "Yuqi Zhang scratched her husband" marked by blue, in CESG-S and CESG-CD. Due to limited space, we only list those three typical cases here. From the result marked by different color, we can conclude that CESG perform significantly better than other four models in The integrity and coherence of storyline.

4.3.2 Overall Performance Comparison. The comparison of our method with other storyline generation methods is presented in Figure 4. The meaning of marks on the figure is same as above. OGO and TTC have different degrees of problem in sequence, marked by green. Their common shortcoming is the integrity of the storyline. Some stages, such as "Jiang's friend sent an article again to support Jiang" cannot be extracted because people participating in the discussion is very few. Sumblr outperform OGO and TTC in sequence. However, it would lead to stage redundancy. (As shown in Fagure4(Sumblr), there are stages with no mark.) The number of stages marked with red in the storyline generated by our model is

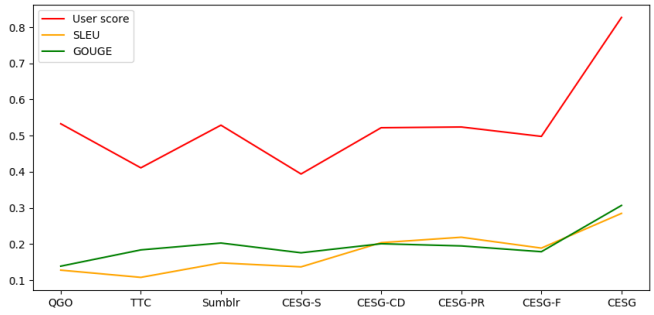

Figure 5: the performance of different evaluation method

obviously more than other three models and stages marked green is less than other three models.

4.4 Quantitative Experiment

The results of quantitative evaluation of the experiments are shown in Table 2. The results, which is the average of all the events in the data sets, show that our model outperforms all the other methods. Figure 5 shows that SLEU is more in line with human evaluation on storyline. In other words, SLEU can quantify the quality of storylines more scientifically .

5 CONCLUSION

In this paper, we propose a novel framework that generate comprehensive storyline automatically for a more user-friendly experience. Specifically, we can generate storyline with the feature of integrity and content coherence, (i.e. understandable for users), from microblogging data related to the specified event. Extensive experiments on real-world Chinese microblogging data demonstrate the effectiveness of the proposed methods in each module and the overall framework.

However, we can still find some shortcomings: 1) The time points of the stages is not accurate. We can only detect the time point of the post when the users release, not the event itself occurs. 2) The generation of the storyline depends on the amount of data at each stage, that is, if there is only one piece of relevant data of each stage, it may be difficult to detect. 3) The storyline is not concise, not clear enough. Users still need to read a long paragraph to get the global information of event. Therefore, we next plan to explore new motheds to make up for the shortcomings mentioned above.

6 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61632004, 61832002, 61672518, 61632115), and the Beijing Municipal Science & Technology Commission (No. Z181100008918012), and the Big Data Application on Improving Government Governance Capabilities National Engineering Laboratory Open Fund Project (Grant No. W-2018028).

REFERENCES

[1] Adrien Guille and Cécile Favre. 2015. Event detection, tracking, and visualization in Twitter: a mention-anomaly-based approach. *Social Network Analysis and Mining* 5 (2015), 1–18.
[2] B. N. Guo, Yi Ouyang, Cheng Zhang, Jiafan Zhang, Zhiwen Yu, Di Wu, and Yu Wang. 2017. CrowdStory: Fine-Grained Event Storyline Generation by Fusion of Multi-Modal Crowdsourced Data. *IMWUT* 1 (2017), 55:1–55:19.
[3] Longtao Huang, Shangwen Lv, Liangjun Zang, Yipeng Su, Jizhong Han, and Songlin Hu. 2018. A Fresh Look at Understanding News Events Evolution.
[4] Chung Jung Lin, Chun Lin, Jingxuan Li, Dingding Wang, Yang Chen, and Tao Li. 2012. Generating event storylines from microblogs. In *CIKM*.
[5] Chin-Yew Lin. 2004. ROUGE: A Package For Automatic Evaluation Of Summaries. In *ACL 2004*.
[6] Chin-Yew Lin and Eduard H. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *HLT-NAACL*.
[7] Fu-Ren Lin and Chia-Hao Liang. 2008. Storyline-based summarization for news topic retrospection. *Decision Support Systems* 45 (2008), 473–490.
[8] Bang Liu, Di Niu, Kunfeng Lai, Linglong Kong, and Yu Xu. 2017. Growing Story Forest Online from Massive Breaking News. *ArXiv* abs/1803.00189 (2017).
[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*.
[10] Hassan Sayyadi and Louiqa Raschid. 2013. A Graph Analytical Approach for Topic Detection. *ACM Trans. Internet Techn.* 13 (2013), 4:1–4:23.
[11] Lidan Shou, Zhenhua Wang, Kan Chen, and Gang Chen. 2013. Sumblr: continuous summarization of evolving tweet streams. In *SIGIR*.
[12] P. Teekens I. Vossen, Tommaso Caselli, and Yiota Kontzopoulou. 2015. Storylines for structuring massive streams of news.
[13] Dingding Wang, Tao Li, and Mitsunori Ogihara. 2012. Generating Pictorial Storylines Via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *AAAI*.