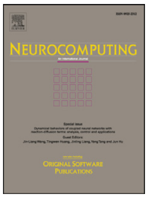
Neurocomputing 397 (2020) 224–243

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Rumor events detection enhanced by encoding sentimental information into time series division and word representations

Zhihong Wang[a], Yi Guo[a,b,c,*]

[a] East China University of Science and Technology, Shanghai, China
[b] National Engineering Laboratory for Big Data Distribution and Exchange Technologies, Shanghai, China
[c] Shanghai Engineering Research Center of Big Data & Internet Audience, Shanghai, China

ARTICLE INFO

Article history:
Received 14 April 2019
Revised 18 January 2020
Accepted 23 January 2020
Available online 28 January 2020

Communicated by Dr. Erik Cambria

Keywords:
Rumor events detection
Sentiment dictionary
Dynamic time series
Cascaded gated recurrent unit
Online social networks

ABSTRACT

Online Social Networks (OSNs) is an ideal place for spreading rumor events as it is convenient in information production and dissemination. Automatically debunking these rumor events is important to pursue and restore the truth. However, it is a challenging task to employ traditional classification approaches for rumor events detection since they rely on hand-crafted features that require daunting manual efforts. Besides, we observe that the various posts of each rumor event will debate its realness over time. Different individuals also have different emotional reactions to events, which will affect others' identification. Thus, this paper firstly employs an automatic construction method to develop a Sentiment Dictionary (SD) to capture the fine-grained human emotional reactions to different events. Secondly, a Two-steps Dynamic Time Series (TsDTS) algorithm, involving the sentimental information in the division process, is elaborated to retain the time-span distribution information of microblog events in a natural manner. At last, a novel two-layer Cascaded Gated Recurrent Unit (CGRU) model based on the SD and the TsDTS algorithm is proposed for rumor events detection, named as SD-TsDTS-CGRU. Experimental results on real datasets from OSNs demonstrate that our proposed SD-TsDTS-CGRU model outperforms the latest rumor events detection algorithms.

1. Introduction

Online Social Networks (OSNs) have increasingly become a powerful tool for journalists but also for ordinary citizens, to gather information and to find out about the latest developments during breaking news stories. However, when social media provides access to an unprecedented source of information, then it also turns to be an important and convenient platform to make the internet rumor events breed and have influence. Not only rumor events set off public incidents, but they also make them worse. Especially the widely spread of rumors about public emergency events, such as natural disasters, accidental disasters, public health incidents, social security incidents, economic crises, etc., will be extremely destructive. This is possible because the highly interconnection and integration between persons in OSNs cause everyone to participate in the process of information generation and dissemination conveniently. Thus, people can pass their true or false messages on to lots of other people at lightning speed [1]. A widely known example is the Fukushima Daiichi nuclear disaster occurred on March 16th of 2011. A rumor event, *"Japan's nuclear radiation will pollute the seawater, causing the salt produced in the future to be inedible"*, was disseminated to millions of persons in a short time period on China's online social network-Sina Weibo [2]. People rushed to buy much more salt than they need in most Chinese cities and rural areas, caused salt panic. This incident of false rumor evidence that it is a highly practical application value to recognize rumor events in an automatic and efficient manner in OSNs.

Most of the previous traditional machine learning-based research work rely on the cues from microblogs themselves, including shallow statistical features (such as content-based features [3–6], account-based features [7], propagation-based features [8–11], etc.) and deep textual features (such as event sentiments [12], event topics [13], event keywords [14], etc.). Among these features, some researchers maintain that sentiments are more effective textual characteristics for rumor distinguishment [12]. However, prior researchers take the sentiment orientation (positive or negative) into consideration alone, without treating event sentiments in much detail. In fact, the sentiments of an event are ultimately about the human emotional reaction to false or true events. According to Vosoughi et al. [15], false rumor events

[*] Corresponding author at: East China University of Science and Technology, Shanghai, China.
E-mail address: guoyi@ecust.edu.cn (Y. Guo).

https://doi.org/10.1016/j.neucom.2020.01.095
0925-2312/© 2020 Published by Elsevier B.V.

inspired fear, disgust, and surprise in replies, whereas true events inspired anticipation, sadness, joy, and trust. Furthermore, the Chinese Book of Rites (Liji)[1] records, humans have seven different sentiments: Happiness, Like, Anger, Sadness, Fear, Disgust, and Surprise. Therefore, these more fine-grained event sentiment categories will be taken into account in this paper to further improve the accuracy of rumor events detection.

In addition, investigators recently have examined the effects of deep neural networks on rumor events detection. In these studies, researchers simply constructed a sequence model with a time-equal-length input [16,17] or a posts-equal-length input [18] to simulate event posts series. However, such approaches not only without fully considered the diffusion pattern of the original microblog, but also didn't take the aggregation degree of the event posts series in the time dimension into all-round consideration.

To alleviate mentioned-above two problems, we firstly propose an automatic construction approach to develop a Sentiment Dictionary (SD) for the sentiments access of event posts, including sentiment words and sentiment emoticons. Furthermore, each entry in the dictionary is with either of two sentiment orientations and one of the seven different sentiment categories. Thereafter, to divide the event posts series in a better manner, we present a Two-steps Dynamic Time Series (TsDTS) division method. In which the first step introduces the theory of domain division of the fuzzy time series model to treat event time spans as a domain, and encode sentimental information into posts series division based on information granule in the second step. On these two bases, an automatic rumor events detection model is proposed in this paper. In summary, our main contributions are as follows:

- Introduce an automatic construction method to develop a Sentiment Dictionary (SD) to capture the fine-grained human emotional reactions to different events, including sentiment words and sentiment emoticons.
- Encode the sentimental information into the process of time-series division and implement a Two-steps Dynamic Time Series (TsDTS) division algorithm for event posts series, which is based on fuzzy clustering and information granule.
- Propose a two-layer cascaded GRU (CGRU) model to capture the variation of contextual and sentimental information each event over time for rumor events detection from microblogs, which is based on the Sentiment Dictionary and TsDTS division algorithm, named SD-TsDTS-CGRU in short.
- Evaluate the effectiveness of SD-TsDTS-CGRU in rumor events detection by comparing with previous state-of-the-art methods on two real and popular social media datasets: Sina Weibo and Twitter.

The reminder of the paper is organized as follows. In Section 2, we give a brief review of related works about rumor detection, time series division and sentiment analysis. Afterward, we introduce the architecture of our rumor events detection model (SD-TsDTS-CGRU), and provides the details of how we construct the SD and the TsDTS division algorithm in Section 3. Then, Section 4 evaluates our methods with dedicated experiments and this paper ends with the conclusion in Section 5.

2. Related work

Rumor events detection problem can be regarded as a classification problem, hence much researches try to solve the problem using handcrafted features and conventional machine learning techniques. Recently, to avoid handcrafted features construction, deep learning techniques are introduced to solve this problem. These machine-learning-based or deep-learning-based detection methods will be summarized in the following Section 2.1. Furthermore, we highlight some methods of rumor events detection improved by time series division (Section 2.2) and sentiment analysis (Section 2.3).

2.1. Rumor events detection methods

Rumor events detection problem can be regarded as a classification problem of machine learning. To train a machine learning algorithm with a set of rumor-like events labeled as Rumor (R) and regular events labeled as Non-Rumor (NR). The idea is to make an algorithm that can learn characteristics of rumor events from the training set so that it can filter out rumor events when encounters new events.

Existing relevant works primarily focus on two aspects including the selection/improvement of classification algorithm and the selection/extraction of rumor events detection features. Yang et al. [3] examined five types of features extracted from the microblogs including content-based features, account-based features, propagation-based features, client-based features and location-based features, and trained a classifier to automatically detect rumors from a mixed set of true and false information. Furthermore, much of the current literature on rumor detection pays particular attention to temporal patterns of the above features. Kwon et al. [6,8] introduced a time-series fitting model based on the temporal properties of post volume. Then Ma et al. [5] extended the model using time series to capture the variation of content-based features over time. In the study of Wu et al. [9] and Ma et al. [10], they primarily focused on the propagation patterns of posts. A graph-kernel based hybrid SVM classifier and a Propagation Tree Kernel (PTK) based classifier were proposed for detecting rumor events from microblogs, respectively. A recent study of ourselves [11] similarly constructed the dynamic time series features over time by introducing the idea of domain division. But overall, most above approaches focus on the shallow statistical features without taking the deep semantic features of rumor events into consideration. Thus, numerous studies have attempted to extract new deep textual features. In the study of [13], the dynamics theory in physics was introduced to model the topic features spreading among the Sina Weibo platform and they proposed a rumor detection method based on the burst topic features. Moreover, to solve the semantic sparse of short posts, [14] introduced a semantic layer between event posts and labels to construct a multi-label bi-term topic model for rumor detection. Besides, Vaghela et al. [19] proposed a method involved the deep features of sentiments to detect rumor events effectively. Similarly, [20] performed a similar series of experiments to show that the event sentiment assisted to identify domain-specific rumors.

In recent years, deep neural networks achieve great performance on many NLP tasks. In rumor events detection field, Ma et al. [16] firstly constructed a Recurrent Neural Network (RNN) modeling the social context information of an event as N time-equal sequences, which could capture the variation of contextual information of relevant posts over time. Chen et al. [18] improved the input sequences of RNN by grouping posts into batches according to a fixed post amount and presented a deep attention model on the basis of RNN to learn selectively temporal hidden representations of sequential posts for identifying rumors. Apart from the previous works, [21] proposed a combination of RNN and variant Auto Encoders (AE) to learn the normal behaviors of individual users for distinguishing rumors as anomalies from other credible microblogs. Besides, Liu et al. [22] presented a rumor detection model based on Convolution Neural Network (CNN), vectorized the rumor events in microblogs and mined the deep features of texts through the learning and training of hidden layer of CNN, which

[1] https://en.wikipedia.org/wiki/Book_of_Rites.

avoided the problem of feature construction. The rumor events detection methods based on deep neural networks adopt continuous vectors to represent the rumor event texts, which overcome the problem of feature sparseness and effectively and have big boosts in accuracy.

2.2. Time series division for rumor events detection

Online Social Networks (OSNs) remodel the method of information transmission, whose openness prompts everyone to be the information resource and propagative channel. When an event especially a rumor event or a hot event broke in OSNs, everyone can participate in the discussion of the event, which will produce plenty of reposts and comments. The posts series of each event in OSNs is a typical time-series data, which is made up of massive amounts of event-related posts, reposts and comments. For such massive time-series data, a lot of segmentation algorithms have been proposed to improve the accuracy of the data mining system. That is, a long time series is divided into several relatively short sub-interval to perform clustering/classification analysis, detecting outliners in time series and other time-series data mining tasks [23]. Algorithms currently used for time series segmentation can be based on the following three categories: limited segments based algorithms [24], segmented error-fitting based algorithms [25,26] and time-series features based algorithms [27]. The divided time-series data has the following advantages for data mining than original [28], (1) high compression on time-series data reduces the data amount and benefits data storage and processing. (2) Reducing the effect of outliners, well keeping the main characteristic of the original time-series data, and significantly improving the data-mining accuracy. (3) The analysis of variation patterns from time-points to time-intervals consistents with the most fields requirements and concerns.

However, to the best of our knowledge, there are only a few researches that divide the microblog event posts series into several intervals to construct a detection model to achieve higher performance in the field of rumor events detection in OSNs. Kwon et al. [6,8] only introduced a time-series fitting model based on the temporal properties of post volume to capture how rumors spread over time. Then Ma et al. [5] extended the model using time series to capture the variation of content-based features over time. Besides, in the latest popular deep learning algorithms, researchers frequently treat each post as an input unit and construct a sequence model (like RNN, LSTM, etc.), in which the time series with a sequence length equal to the number of event posts [62]. However, there is a large difference in the number of microblog posts between different types of events. That is, it is not practical to deal with each post individually in the large number scale. Therefore, when a sequence model is uniformly constructed, more input units are required to cover all types of events with only one output unit. Hence, some researchers present different methods to construct sequence models by batching posts into time interval and treating them as a single unit in a time series. These methods can be divided into two parts: time-equal-length based time series division (TETS) [16,17] and posts-equal-length based time series division (PETS) [18]. The main idea of TETS is to divide the event series into an equal length of time, which can ensure the model with an optimal input sequence. However, with the fixed time length it will result in an excessive number of microblog posts in part of the time series and a few numbers of posts in some others. Therefore, some researchers propose the method with fixed posts length, the main idea of PETS, which guarantees a reasonable number of time series divisions and also ensure the reasonable number of posts in each time series.

Although the above-mentioned methods have considered the reasonable number of time-series partitions and reduced the computational cost, they rarely consider the distribution characteristics of event posts themselves in the time dimension, including the distribution of event time, posts, sentiments, etc. Intuitively, the length of sub-interval should be short in areas with dense event posts and the length of the sub-interval with sparse data should be longer. In this case, the detection models have the ability to capture the variation of contextual information and the evolution patterns of events. This is the basic inspiration for constructing variable-length microblog event posts series as the inputs of rumor events detection models in this paper.

2.3. Sentiment analysis for rumor events detection

Sentiment analysis [29] is a “suitcase” research problem that requires talking many NLP sub-tasks [30], including aspect extraction [31,32], subjectivity detection [33], concept extraction [34], and sarcasm detection [35,36]. Along with the Internet as the core information technology’s development, the OSNs continuously produces large quantities of texts with emotions. Hence, sentiment analysis can be used to fully mine these texts to help research institutions, information consultancies, and governments master the dynamics of public opinions. For examples, understanding the view of some netizens contributes to establishing intelligent emergence early-warning systems [37] and perfecting the network supervision [38]. Based on the sentiment analysis of product reviews [31,39] or hotel reviews [40], we can offer valuable clues for product improvements. Moreover, sentiment analysis is also applied to research in the fields of psychology [41], sociology, financial prediction [42], etc. Sentiment analysis, hence, is key for the advancement of artificial intelligence (AI) and all the research fields that stem from it [43]. For now, the main sentiment analysis methods contain the sentiment dictionary-based approach and the machine learning-based approach. While the former method is very simple as well as convenience, it is not advisable as it may result in a different convoluted construct process for different sentiment dictionaries in different fields [44]. Of another machine learning-based approach, sentiment analysis is treated as either a binary (−1 or 1) or multiple class classification problems. Traditional machine learning aims at manual construction of features that could distinguish sentiment of text out of different domain knowledge. Furthermore, as the deep neural networks have developed, recent machine learning turns to adopt pre-trained representation [45,46] and build neural networks (such as CNN [32], RNN [47], LSTM [48], attentive LSTM [49], attention network [50], etc.) to automatically learn capable classification features.

In the field of rumor events detection, sentiment analysis has been introduced to obtain deep semantic features to improve detection accuracy. Ghanem et al. [51] argued that false information has different emotional patterns in each of its types, and emotions play a key role in deceiving the reader. Based on that, they proposed an LSTM neural network model that is emotionally-infused to detect false news. Mao et al. [12] used the ensemble classifiers to detect rumor events, which involved the previous shallow statistical features and the deep feature of sentiment orientations. The authors reported that the sentiment orientations of microblogs were the most effective, enhanced the detection accuracy and F1 of 3.9% and 4.6%, respectively. In a similar vein, Li [52] used the emotion dictionary to analyze the sentiment tendency of microblog comments to get the emotion characteristic and put forward a sentiment-based hybrid kernel SVM (SHSVM) classifier to complete the identification of microblog rumors. Likewise, Zu et al. [53] also found that the features of emotional tendencies of microblog comments brought a considerable improvement (about 4% both on precision, recall and F1) for rumor events detection. Besides, Sivasangari et al. [54] proposed a novel VADER sentiment analysis, which was not only considering the sentiment category, but also considered the intensity (strength) of the text using the rule-based heuristics method with the value for each and every word in the text (tweets). Then they found a sentiment lexicon score value using VADER for the scraped dataset to segregate the rumor with greater accuracy.

Above all, various studies have assessed the efficacy of the deep features of event sentiments and achieved gratifying performance. But we argue that the current better-performing sentiment features of rumor detection only consider the sentiment orientation (Positive or Negative) without any more fine-grained sentiment categories (such as Happiness, Like, Anger, Sadness, etc.) of an event. This is also the ideological origin of the methods that this paper attempts to explore and implement.

3. Methods and models for rumor events detection

This section presents the details of our rumor events detection model for classifying microblog events into rumor or non-rumor, including the overall structure of our proposed model, construction of the sentiment dictionary and variable-length microblog event posts series.

3.1. Problem statement

Individual social posts contain very limited content because of their nature of shortness in content. On the other hand, a news event is generally associated with a number of posts that are relevant to the event. These relevant posts regarding the news event can be easily retrieved to describe the central content more faithfully. Hence, this paper focuses on rumor detection at an aggregated level instead of identifying each single post. That is to say, we concentrate in detecting rumors on event-level wherein sequential posts related to the same topics are batched together to constitute an event, and our model determines whether the event is a rumor or not.

Let $E = E_i$ denotes a set of given events, where each event $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$ consists of all relevant posts $p_{i,j}$ at timestamp $t_{i,j}$, and the task is to determine whether each of these events, E_i, is a rumor or a non-rumor by assigning a label from $Y = \{R, NR\}$.

3.2. Overview of the model

Rumor events in online social networks are typical time-series data. While the RNN models are popular neural networks used for processing sequence data and have achieved good performance in many NLP tasks. Among them, the GRU achieves better results in rumor events detection [16]. Based on the results of [16], a two-layer Cascaded GRU (CGRU) model is adopted as the basic module in this paper. Besides, a Sentiment Dictionary (SD) is constructed to capture the sentiment categories changes of each event and a Two-steps Dynamic Time Series (TsDTS) algorithm is designed to naturally fit the time intervals for each event's posts. The following sections will detail each of these methods. The architecture of our proposed model is shown in Fig. 1.

As can be seen from the figure above, the first layer is the original posts of each microblog event, which are divided into N intervals by a TsDTS algorithm in the second layer. The third layer, an input layer of our cascaded GRU model. The inputs of one GRU model is the top-k words in the vocabulary (denoted as x_t^w), and the input of another GRU model is consist of sentiment words and sentiment emoticons in SD (denoted as x_t^s). To reduce the complexity, we add an embedding layer between the input and hidden layers (the fifth layer in Fig. 1) so that the overall scale of parameters becomes much smaller. The embedding layer is given as Eq. (1). And the hidden layers of the two cascaded GRU models are calculated by Eq. (2) and Eq. (3), respectively. Finally, the results of the cascaded GRU model are concatenated by a dense layer, and the output layer uses the "softmax" output function to determine whether the event is a rumor or not (See Eq. 4).

$$\begin{aligned} x_e^w &= x_t^w E \\ x_e^s &= x_t^s E = (x_t^{sw} \oplus x_t^{se})E \end{aligned} \tag{1}$$

$$h_t^{w,(2)} = GRU(GRU(x_e^w)) \tag{2}$$

$$h_s^{s,(2)} = GRU(GRU(x_e^s)) \tag{3}$$

$$\begin{aligned} z_o &= W_o \cdot [h_t^{w,(2)}, h_t^{s,(2)}] + b_o \\ y &= softmax(z_o) \end{aligned} \tag{4}$$

where E is the word embedding weight matrix. Moreover, we learn this embedding matrix ourselves from Sina Weibo for Chinese rumor events detection and directly use the GloVe Twitter pre-training matrix for English[2]. $h_t^{w,(2)}$ and $h_t^{s,(2)}$ are the hidden layer outputs of the two-layer GRU model of words' information and sentimental information, respectively. W_o are the weight connections of the dense layer and b_o is the bias. y is the probability to demonstrate an event is a rumor or not.

3.3. Construction of the sentiment dictionary

Microblog text contains richer internet slangs than traditional news or online reviews, such as "坑爹" (keng-die/cheat someone), "2333" (laughter), "BS" (bi-shi/to despise) etc. Furthermore, more people enjoy expressing their attitudes by elaborate emoticons in microblog. Most of these two entries are the leading carriers of human sentiments, which are essential for identifying rumor events. In this section, we will explore the construction of Sentiment Dictionary (SD) about Chinese microblog. It mainly includes *preprocesses of microblog text, construction of Sentiment Word Dictionary (SWD)* and *construction of Sentiment Emoticon Dictionary (SED)*.

3.3.1. Pre-processing of microblog texts

Microblog texts have numerous characteristics that ordinary texts do not take. The most obvious is that microblog texts often contain some pictures, Web links, mention someone's symbol "@", topic symbols "##", "#" or "【 】" and more. Although such information brings many colorful expressions to microblog texts, it also makes some researches tougher. Hence, it is essential to preprocess microblog texts to facilitate the researches. Different previous studies have different pretreatment for microblogs. This paper is in view of sentiment dictionary construction for microblog texts, and the pretreatments include the aspects as follows.

(1) *Filtering multimedia.* Because the limitation of microblog texts, people sometimes supplement the microblog texts with some additional information like the Web link "https", gif, video, image, emoticon and other forms. The information may have a certain impact on microblogs, but it is always complicated and confused, and it is also useless for the construction of the sentiment dictionary. Thus, they will be filtered in this paper.

(2) *Filtering "@ + username".* The symbol "@ + username" is used to get someone's attention in Microblogs. While, for construction of the sentiment dictionary it does not perform a real function, so we will filter them too.

(3) *Filtering "# + topic + #", "# + topic" or "【+ topic +】".* This part information indicates the topic of each current post. However, the microblog texts of an event are collected and sorted according

[2] glove.twitter.27B: http://nlp.stanford.edu/data/glove.twitter.27B.zip.

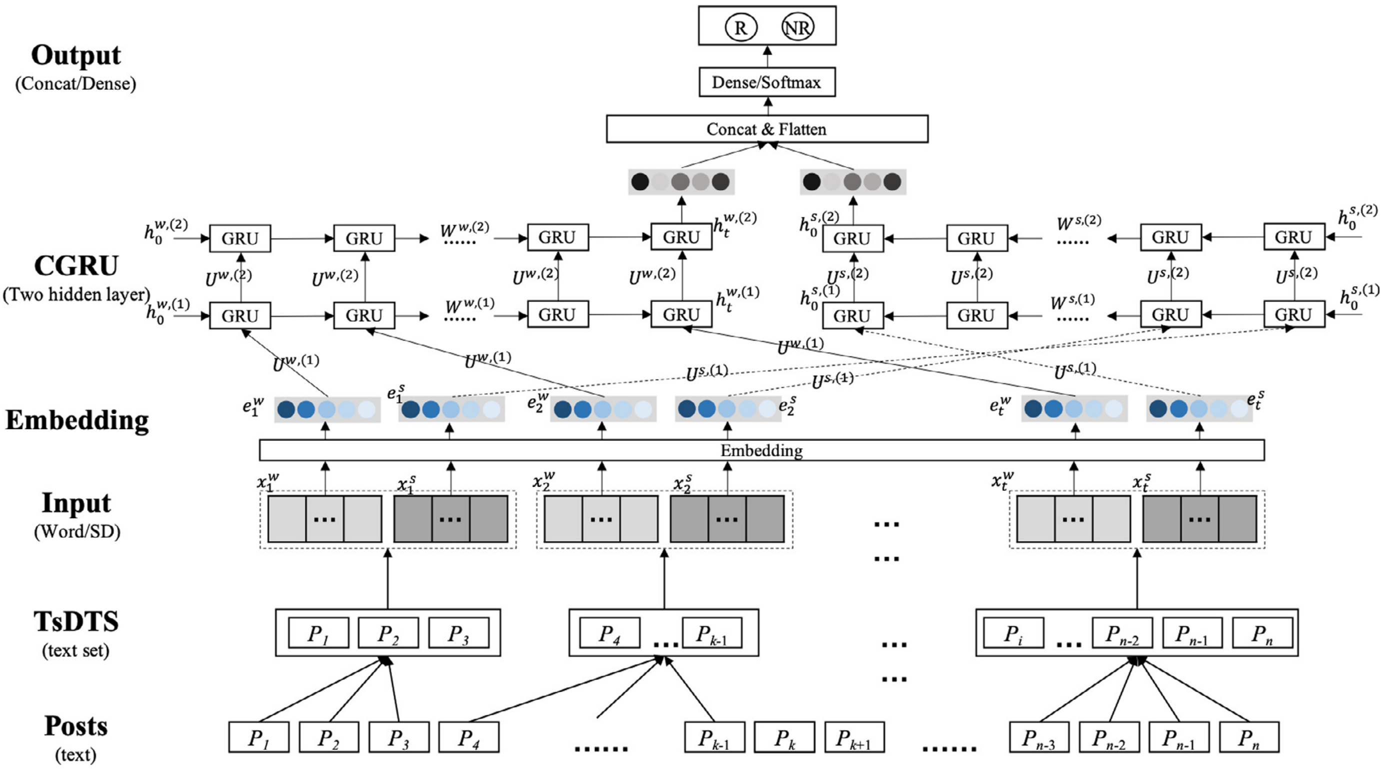


Fig. 1. The architecture of rumor events detection model.

to the same topic, hence they should be ignored and filtered out immediately.

(4) *Converting traditional Chinese to simplified and English to lowercase.* Due to the mixture of traditional and simplified Chinese or English in lowercase and uppercase, these characters will be automatically converted into simplified Chinese and English in lowercase, respectively.

The particularity of microblog texts effect the requirement of special pre-processes in sentiment dictionary construction. Reasonable pre-processes will improve the quality of microblog texts processing, thereby further accelerate the processing of subsequent word segmentation and stop-word removal.

3.3.2. *Construction of the sentiment word dictionary*

Sentiment Word Dictionary (SWD) in this paper includes two parts: Basic Sentiment Word Dictionary (BSWD) and Extension Sentiment Word Dictionary (ESWD). Among them, the BSWD for Chinese is the Affective Lexicon Ontology (ALO) [55] constructed by Dalian University of Technology with seven categories of sentiment words, including Happiness, Like, Anger, Sadness, Fear, Disgust, and Surprise. Each entry in the dictionary is with one of the seven sentiment categories. Moreover, all entries are also labeled with an integer to indicate their sentiment orientations, including negative, positive or neutral. For English, the BSWD is the manual annotated NRC Emotion Lexicon [56], which is a list of English words and their associations with eight basic emotions (Anger, Fear, Anticipation, Trust, Surprise, Sadness, Joy, and Disgust) and two sentiments (negative and positive). Noted that the eight basic emotions of NRC will be mapped to the seven sentiment categories of ALO in this paper, seen in Table 1.

All in all, the BSWD is consist of sentiment words in written form, which can identify a sentiment word in the formal expression in microblog texts. But a sentiment word can be expressed in different styles, especially in the colloquial microblog texts. This leads to lower recall performance for the construction of sentiment dictionary. Hence the ESWD is constructed to complete the sentiment word dictionary to improve recall performance. This section mainly details the construction of ESWD.

Table 1
The mapping between NRC emotions and sentiment categories.

NRC emotions	Sentiment category	Examples
Joy	Happiness	Laugh, Rapture, Rave
Anticipation	like	Accolade, Achievement
Trust		Befriend, Believed
Anger	Anger	Preclude, Prejudice, Vengeance
Sadness	Sadness	Tolerate, Tomb, Aggravating
Fear	Fear	Smuggle, Abduction, Snare, Confine
Disgust	Disgust	Aberration, Entrails
Surprise	Surprise	Evanescence, Gape, Inexplicable

Given a sentiment category, its sentiment words usually have similar semantics. This motivates us to use word embedding technology in deep learning to discover the synonyms of seed sentiment words for building an extension sentiment word dictionary. The specific methods to achieve the following,

(1) Sentiment words extension with similar semantics

Prior to sentiment words extension, the first important task is how to represent a word in semantics. Word2vec[3], a well-known word embedding tool, is trained on microblog texts to represented each word as a vector with a specified number of dimensions, such as 100 or 200 etc. The relatedness of the two words can be obtained by calculating the similarity between the two corresponding vectors. Thus, the initial ESWD is consist of the words from microblog texts, whose similarity with one of the words in BSWD is greater than a threshold.

[3] https://code.google.com/archive/p/word2vec/

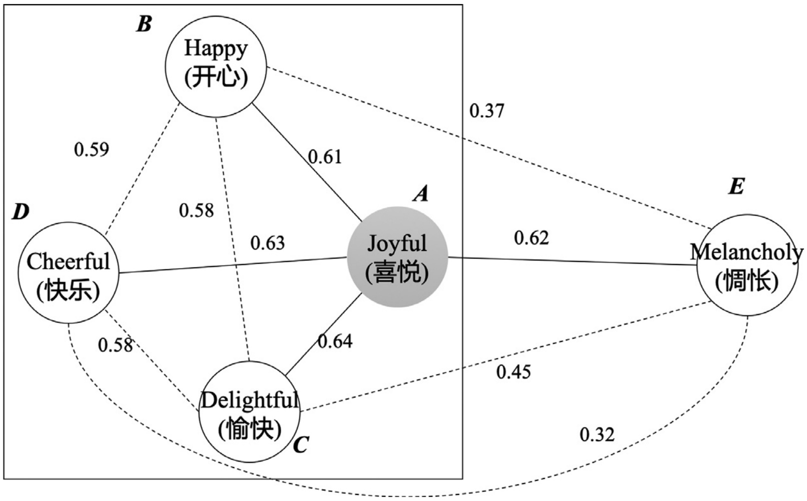


Fig. 2. Example of similar words for “Joyful (喜悦)” .

(2) Irrelevant words removal

In theory, the words in initial ESWD have high similarity in sentiment semantics with a seed sentiment word in BSWD. However, we observe from experiments that most of the words in initial ESWD have similar sentiment semantics, but there are still some irrelevant words and even antonyms of the word. For example, given a seed word “Joyful (喜悦)” in Chinese, word2vec discovers some similar words (See Fig. 2), such as “Happy (开心)”, “Cheerful (快乐)”, “Delightful (愉快)”, “Melancholy (惆怅)”, etc. The value on each edge in Fig. 2 indicates the similarity between the two words. The larger the value, the higher the similarity. It is obvious that “Melancholy (惆怅)” is similar with “Joyful (喜悦)” in semantics but opposite in sentiment, which should be deleted. The main reason for this is that word2vec is based on the hypothesis that the words in similar contexts have similar meanings. In the training corpus the word “Melancholy (惆怅)” have similar contexts to the seed word “Joyful (喜悦)”.

To remove the irrelevant words from the initial ESWD, we present a majority voting algorithm based on triangle relations. The fundamental idea of this algorithm is that if two words are similar to the seed word in sentiment, they are also similar to each other. That is to say, in high-dimensional word vector space, three points representing similar sentiment form a triangle, where the weight of each edge is greater than a threshold. Moreover, there are multiple candidate words with similar sentiments for each seed word and they will form multiple triangles. For a candidate word, if the weight of the edge between it and half of other candidates is not greater than the threshold (The threshold θ is set to 0.5 in this paper, discussed in the subsequent experiments), this irrelevant candidate word will be removed according to the triangle relations. In Fig. 2, the seed sentiment word “Joy (喜悦)” (A) has four similar words, represented by B, C, D and E. Each edge of triangles, △ABC、△ABD and △ADC, is greater than the threshold, but the edges BE of △ABE, CE of △ACE and DE of △ADE are less than the threshold, which means that the sentiment semantic of E is different from other words. The distance between “Joy (喜悦)” and “Melancholy (惆怅)” is short only because the two words appear in similar contexts in microblog texts. Thus “Melancholy (惆怅)” is considered as an irrelevant word and removed from the ESWD.

(3) Sentiment orientations and categories identification

To further complete the ESWD for event sentiment analysis, this section will calculate the sentiment orientations and categories for each extended word. Since a word in ESWD is extended by words in BSWD according to the word2vec model, so each extended word ESW_i can be represented by a set of words in BSWD. That is, $ESW_i = \{<BSW_{i1}, S_{i1}>, <BSW_{i2}, S_{i2}>, \ldots, <BSW_{in}, S_{in}>\}$, where BSW_{ij} is the jth basic sentiment word similar to ESW_i, and S_{ij} is the similarity. Thus, the sentiment orientation of each extended word ESW_i in this paper is calculated by the equation below.

$$O(ESW_i) = \begin{cases} 1, & \text{if } \sum_{j=1}^{n} S_{ij} O(BSW_{ij}) > 0 \\ -1, & \text{if } \sum_{j=1}^{n} S_{ij} O(BSW_{ij}) < 0 \end{cases} \tag{5}$$

And the sentiment category is calculated by the following equation.

$$C(ESW_i) = \{C| \max_{C} \sum_{j=1}^{n} \{1 * S_{ij} | C(BSW_{ij} = C)\}\} \tag{6}$$
$$s.t. C \in \{Hapiness, Like, Anger, Sadness, Fear, Disgust, Surprise\}$$

Where $O(BSW_{ij})$ is the sentiment orientation of basic sentiment word BSW_{ij} (1 is positive, -1 is negative). Then the sentiment orientation of ESW_i corresponds to the weighted accumulation of the orientations of all basic sentiment words $\{BSW_{ij}\}_{j=1}^{n}$, expressed as $O(ESW_i)$. $C(BSW_{ij})$ is the sentiment category of the basic sentiment word BSW_{ij}, and $C(ESW_i)$ is the sentiment category of the extended sentiment word ESW_i. A similar approach with $O(ESW_i)$, the sentiment category of ESW_i is one of the seven sentiment categories with the largest weighted accumulation, represented by $C(ESW_i)$.

3.3.3. Construction of the sentiment emoticon dictionary

Scott Elliott Fahlman is credited with originating the first smiley emoticon [57]. He proposed the use of :-) and :-(to help people to distinguish serious posts from jokes on a message board at Carnegie Mellon. An emoticon is a textual portrayal of writers' mood or facial expression, which can present their sentiments to others in a direct manner. Similarly, in online social media, emoticons are also the most direct and convenient form for individual sentiments expression. Therefore, this paper will construct a Sentiment Emoticon Dictionary (SED) to capture the changes of event sentiments over time. The construction includes the following steps.

(1) Emoticons collection

Regular expressions are used to extract emoticons from microblog texts. Then the emoticons that used with high frequency are selected to build the emoticon dictionary in this paper. There are 200 emoticons, and around 97.% of the total usage.

(2) Sentiment orientation and category identification

In microblog texts, part of emoticons is with bright sentiment information, for example, 😆 is obviously positive and belongs to “Happiness” category. Others have no definitive sentiments, such as 🌿, 🤖, etc. And still others’ sentiments will vary according to persons’ use case, for example, the emoticon 🤔 sometimes means “Good Bye”, but in some case, it also expresses “Don’t want to see you again”, that is negative and belongs to “Disgust” category. That is, the sentiment information of emoticons is complicated and changeable. Thus, it is necessary to design a dynamic approach to identify sentiment orientation and category automatically.

From the microblog texts, we observe that microblog users prefer to strengthen the expression of their sentiments with some emoticons. For example, “AH, AH... I AM SO SAD 😢😢😢”. In this post, “SAD” is a negative sentiment word, belongs to “Sadness” category. And the emoticon 😢 is used to strengthen the user’s mood, which is also negative and belongs to “Sadness” category. That is, sentiment words and emoticons with the same sentiment orientation and category tend to appear simultaneously. Thus, the co-occurrence relationships between emoticons and sentiment words are employed in this paper to identify the emoticons’ sentiment orientations and categories.

Sentiment Orientation. For each sentiment emoticon SE_i, the co-occurrent sentiment words are represented as $\{SW_{i1}, SW_{i2}, \ldots, SW_{im}\}$. Therefore, the sentiment orientation of emoticon is calculated as follows.

$$O(SE_i) = \begin{cases} 1, & \text{if } \sum_{j=1}^{n} O(SW_{ij}) > 0 \\ -1, & \text{if } \sum_{j=1}^{n} O(SW_{ij}) < 0 \end{cases} \tag{7}$$

Where SW_{ij} is the jth co-occurrent sentiment word with SE_i, and $O(SW_{ij})$ is the sentiment orientation of SW_{ij} (1 is positive, −1 is negative).

Sentiment Category. This paper introduces the Emoticon Sentiment Category Tendency (ESCT) to evaluate the tendency of each emoticon for different sentiment categories, which is affected by two factors, including Emoticon Sentiment Saliency (ESS) and Emoticon Sentiment Relevance (ESR).

ESS represents the intensity of an emoticon expressing different sentiment categories. If an emoticon is always used to express a specific sentiment category, that is to say, this emoticon has a more significant tendency for such sentiment category. Thus, ESS is given as the the following equation.

$$ESS_{ij} = \frac{CoCount(SE_i, CSW_j)}{CoCount(SE, CSW_j)} \tag{8}$$

Where ESS_{ij} indicates the ith emoticon tends to the jth sentiment category. CSW_j is the jth sentiment category. $CoCount(SE_i, CSW_j)$ is the co-occurrent frequency of the ith emoticon and the jth sentiment category, which is calculated by the co-occurrence of SE_i and sentiment words. Moreover, $CoCount(SE, CSW_j)$ is the co-occurrent frequency of all emoticons and the jth sentiment category.

Besides, ESR represents the ability of an emoticon to distinguish different sentiment categories. It means that if an emoticon is co-occurring with sentiment words in most sentiment categories, the emoticon has a lower sentiment category relevance. ESR is given as Eq. 9 in this paper.

$$ESR_i = \log \frac{CoCount(SE, CSW)}{1 + CoCount(SE_i, CSW)} \tag{9}$$

Where ESR_i is the relevance between the ith emoticon and sentiment category. $CoCount(SE, CSW)$ and $CoCount(SE_i, CSW)$ is the co-occurrent frequency of all sentiment categories and all emoticons or the ith emoticon, respectively.

Hence, the trend value of the ith emoticon belonging to the jth sentiment category, represented as $ESCT_{ij}$, is calculated by the following equation.

$$ESCT_{ij} = ESS_{ij} * ESR_i \tag{10}$$

At last, according to Eq. (11), each emoticon in this paper will have seven different trend values for seven different sentiment categories. Then the sentiment category with maximum trend value are selected as each emoticons’ sentiment category.

3.4. Constructing variable-Length microblog event posts series

In this section, a novel division algorithm, named Two-steps Dynamic Time Series (TsDTS), is proposed for event posts series, which is based on fuzzy clustering and information granule. In the first step, we introduce the theory of domain division of the fuzzy time series model, and present an initial division algorithm for the event posts series based on fuzzy clustering algorithm by considering the event time span as a domain. It can be observed that the time distribution information of events is taken a full consideration in the initial division. According to the degree of membership of the first step, a fuzzy segmentation of the time series can be obtained, not the exact time-division point. Thus, this paper proposes the second-step division algorithm based on information granule to further clarify the exact time-division point and simultaneously make the data in the segmented event subsequence have the same statistical properties. That is to say, the distribution characteristics of event posts themselves in the time dimension can be considered in the second-step division by introducing information granule, which brings the different segmented subsequences to own the trend characteristics of event posts

3.4.1. Notation definition

For each event $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$, the observation dataset of the time series is $T_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,n_i}\}$. $T_i^{\max} = max\{t_{i,j} | t_{i,j} \in T_i\}$ and $T_i^{\min} = min\{t_{i,j} | t_{i,j} \in T_i\}$ are the maximum and minimum values of T_i. $U_i = [U_i^{start}, U_i^{end}] = [T_i^{\min} - l_1, T_i^{\max} + l_2]$ is the domain of time series T_i, where l_1 and l_2 are pruning operators. The main aim of Section 3.4 is to divide the time series of each event into K subintervals with different lengths ($k \geq 2$). In the following subsections, we will detail the proposed Two-steps Dynamic Time Series (TsDTS) division algorithm for event posts series.

3.4.2. First-step: posts series division based on fuzzy clustering

The fuzzing clusetring algorithm adopted in this paper is the Fuzzy C-Means (FCM) algorithm, proposed by Bezdek [58] in 1981, which is the most widely used fuzzy clustering algorithm. The FCM algorithm attempts to partition M elements into a collection of N fuzzy clusters with respect to some given criterion. The objective function and constraints are as follows.

$$J(U,V) = \sum_{i=1}^{M} \sum_{n=1}^{N} (u_{in})^{\beta} d^2(x_i, v_n)$$
$$s.t. \begin{cases} 0 \leq u_{in} \leq 1, & \forall i, n \\ 0 \leq \sum_{i=1}^{M} u_{in} \leq M, & \forall n \\ \sum_{n=1}^{N} u_{in} = 1, & \forall i \end{cases} \tag{11}$$

where $\beta \geq 1$ is the hyper-parameter that controls how fuzzy the cluster will be. The higher it is, the fuzzier the cluster will be in the end. $d(x_i, v_n)$ is t e distance between the ith data and the nth center of cluster. u_{in} is the degree to which element, x_i, belongs to cluster v_n.

To find the extremum of the objective function with constraints, the Lagrange factor is introduced to reconstruct a new objective function.

$$J_\lambda(U,V) = \sum_{i=1}^{M}\sum_{n=1}^{N}(u_{in})^\beta d^2(x_i, v_n) + \lambda\left(\sum_{n=1}^{N} u_{in} - 1\right) \tag{12}$$

And the optimization conditions for the extremum of the objective function are as follows.

$$\begin{aligned} \frac{\partial J_\lambda}{\partial \lambda} &= \sum_{n=1}^{N} u_{in} - 1 = 0 \\ \frac{\partial J_\lambda}{\partial u_{in}} &= \sum_{n=1}^{N} \beta (u_{in})^{\beta-1} d^2(x_i, v_n) - \lambda = 0 \\ \frac{\partial J_\lambda}{\partial v_n} &= \sum_{n=1}^{N}(u_{in})^\beta x_i - v_n \sum_{i=1}^{N}(u_{in})^\beta = 0 \end{aligned} \tag{13}$$

Hence, the formulas for the degree of membership and cluster center are as follows.

$$u_{in} = \frac{1}{\sum_{n'=1}^{N}\left(\frac{d(x_i, v_n)}{d(x_i, v_{n'})}\right)^{\frac{2}{\beta-1}}}, \; v_n = \frac{\sum_{i=1}^{N}(u_{in})^\beta x_i}{\sum_{i=1}^{N}(u_{in})^\beta} \tag{14}$$

Besides, in the absence of experimentation or domain knowledge, β is commonly set to 2. The condition for stopping the FCM algorithm is that the number of iterations reaches up to 100, or the coefficients' change between two iterations is no more than $1 * 10^{-5}$.

Therefore, in the first-step division, the posts series of each event E_i are divided into N sub-intervals with different lengths, where $N = \lfloor \frac{K}{2} \rfloor$. More specifically, we choose the middle point of two adjacent cluster centers as the critical point of the time span based on the cluster centers calculated by the FCM algorithm. That is to say, for the cluster centers $(v_{i,1} < v_{i,2} < \cdots < v_{i,N})$, $m_{i,n} = \frac{v_{i,n}+v_{i,n+1}}{2} (i = 1, 2, \ldots, N-1)$ is the middle point of two adjacent cluster centers of vent E_i. After that, the time intervals $T_{i,1}, T_{i,2}, \ldots, T_{i,N}$ are obtained, where N is the number of cluster centers. And the sub-intervals of each event E_i in the first-step division are: $T_{i,1} = \{t_{i,j} \in T_i | T_i^{\min} \le t_{i,j} \le m_{i,1}\}, T_{i,2} = \{t_{i,j} \in T_i | m_{i,1} \le t_{i,j} \le m_{i,2}\}, T_{i,3} = \{t_{i,j} \in T_i | m_{i,2} \le t_{i,j} \le m_{i,3}\}, \ldots, T_{i,N} = \{t_{i,j} \in T_i | m_{i,N-1} \le t_{i,j} \le T_i^{\max}\}$.

3.4.3. Second-step: posts series re-division based on information granule

The concept of information granule was proposed by Zadeh in 1979 [59]. For the one-dimensional time series dataset $T_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,n_i}\}$, the information granule is an interval on R, which is noted as $\Omega_i = [a_i, b_i]$, where a_i, $b_i \in R$ are the lower and upper boundary of information granule, respectively. The purpose of information granules is to obtain appropriate information granularity and semantic interpretation of this interval.

For appropriate information granularity, the more data falling into the interval, the more information the interval Ω_i can carry. Generally, the appropriate granularity is measured quantitatively by the amount of data falling into the interval (denoted as $Card\{y_i | y_i \in \Omega_i\}$), that is, by using the incremental function of the dataset cardinality (denoted as $f_1(Card\{y_i | y_i \in \Omega_i\})$). In this paper, the number of posts in each interval is used to compute the information granule for each event posts series, denoted as $f_1(Card\{p_{i,j} | p_{i,j} \in \Omega_i\})$. What's more, the fine-grained sentiment categories of an event are important for rumor events detection, which have been evaluated in [15]. Therefore, in addition to the number of posts, this paper also takes the number of sentiment words of each fine-grained sentiment category in each interval Ω_i into consideration. Hence, the calculation function of information granularity in this paper is denoted as follows:

$$\begin{aligned} & y_1 = f_1(Card\{p_{i,j} | p_{i,j} \in \Omega\} + \sum_{c \in C} Card\{SD_i^C | SD_i^C \in \Omega_i\}) \\ & s.t. C \in \{Hapiness, Like, Anger, Sadness, Fear, Disgust, Surprise\} \end{aligned} \tag{15}$$

where $Card\{p_{i,j} | p_{i,j} \in \Omega\}$ is the number of posts of an event E_i in the interval Ω_i. Similarly, $Card\{SD_i^C | SD_i^C \in \Omega_i\}$ is the number of sentiment words of cth sentiment category in this interval.

Besides, appropriate semantic information is measured by the interval length of the information granule, denoted as $m(\Omega_i) = |b_i - a_i|$. The smaller the length of an interval is, the more specific the semantics can be described, which is also in line with the persons' common understanding. More generally, the decremental function $f_2(m(\Omega_i))$ is used to measure the semantic information of Ω_i. Hence, the calculation function of semantic information in this paper is denoted as follows:

$$y_2 = f_2(m(\Omega_i)) \tag{16}$$

Above all, the lower $(a_{i,n})$ and upper $(b_{i,n})$ boundaries of the information granule in each sub-interval $(T_{i,n})$ of the event E_i are calculated by maximizing the product of these two functions. The equations are as follows.

$$\begin{aligned} V(a_{i,n}) &= y_1(a_{i,n}) * y_2(a_{i,n}) \\ &= f_1(Card\{p_{i,j} | p_{i,j} \in [a_{i,n}, med(T_{i,n})]\} \\ &\quad + \sum_{c \in C} Card\{SD_{i,n}^c | SD_{i,n}^c \in [a_{i,n}, med(T_{i,n})]\}) \\ &\quad * f_2(|med(T_{i,n}) - a_{i,n}|) \end{aligned} \tag{17}$$

$$\begin{aligned} V(b_{i,n}) &= y_1(b_{i,n}) * y_2(b_{i,n}) \\ &= f_1(Card\{p_{i,j} | p_{i,j} \in [med(T_{i,n}), b_{i,n}]\} \\ &\quad + \sum_{c \in C} Card\{SD_{i,n}^c | SD_{i,n}^c \in [med(T_{i,n}), b_{i,n}]\}) \\ &\quad * f_2(|med(T_{i,n}) - b_{i,n}|) \end{aligned} \tag{18}$$

where $med(T_{i,n})$ is the average of every sub-interval $T_{i,n}$, which is represented by the cluster center $(v_{i,\,n})$ from the first-step division. Take $V(b_{i,opt}) = \max_{b_i \ge med(T_{i,n})} V(b_{i,n})$, then the optimal upper bound $b_{i,opt}$ is the upper bound of the information granule in this interval $T_{i,n}$. In the same way, the lower bound of the information granule in this interval is $a_{i,n} = a_{i,opt}$, where $V(a_{i,opt}) = \max_{a_i \le med(T_{i,n})} V(a_{i,n})$. For the convenience of computing, the two functions are the same as [60] and as follows.

$$\begin{aligned} f_1(x) &= x \\ f_2(x) &= e^{-x} \end{aligned} \tag{19}$$

Therefore, according to Eqs. (17) and (18), the optimal lower $(a_{i,n})$ and upper $(b_{i,n})$ boundaries of the sub-intervals of time series obtained by the first-step division algorithm can be calculated. Then, we can get K sub-intervals $(u_{i,1}, u_{i,2}, \ldots, u_{i,K})$ of different lengths for each event E_i, and the equations of the second-step division algorithm based on information granule are as follows.

When K is odd,

$$u_{i,k}=\begin{cases} u_{i,1}=[U_i^{start}, med(T_{i,1})] \\ u_{i,2n}=\left[med(T_{i,n}), \frac{b_{i,n}+a_{i,n+1}}{2}\right], n=1,2,3,\ldots,\frac{K-2}{2} \\ u_{i,2n+1}=\left[\frac{b_{i,n}+a_{i,n+1}}{2}, med(T_{i,n+1})\right], n=1,2,3,\ldots,\frac{K-2}{2} \\ u_{i,K-1}=\left[med(T_{i,\frac{K-1}{2}}), \frac{med(T_{i,\frac{K-1}{2}})+U_i^{end}}{2}\right] \\ u_{i,K}=\left[\frac{med(T_{i,\frac{K-1}{2}})+U_i^{end}}{2}, U_i^{end}\right] \end{cases} \tag{20}$$

When K is even,

$$u_{i,k}=\begin{cases} u_{i,1}=[U_i^{start}, med(T_{i,1})] \\ u_{i,2n}=\left[med(T_{i,n}), \frac{b_{i,n}+a_{i,n+1}}{2}\right], n=1,2,3,\ldots,\frac{K-1}{2} \\ u_{i,2n+1}=\left[\frac{b_{i,n}+a_{i,n+1}}{2}, med(T_{i,n+1})\right], n=1,2,3,\ldots,\frac{K-1}{2} \\ u_{i,K}=\left[med(T_{\frac{K}{2}}), U_i^{end}\right] \end{cases} \tag{21}$$

3.4.4. Tsdts division algorithm for event posts series

After the two steps division in Sections 3.4.2 and 3.4.3, each event can obtain K unequal intervals. These divided sub-intervals not only take the time information of each different event into consideration, but also consider the relationship between the sub-interval and the data distribution characteristics, such as the number of posts and sentiments. Specifically, the Two-steps Dynamic Time Series (TsDTS) division algorithm for event posts series is designed as follows. According to the distribution of each event time series, this paper firstly clusters the event posts series into N intervals based on the FCM algorithm. Then, the information granule of each interval is calculated to obtain the optimal upper and lower boundaries for these intervals. At last, the intervals obtained by the first-step division is adjusted by the information granule of each interval to finally obtain K unequal-length divisions of each event posts series. The pseudo-code of TsDTS, based on FCM algorithm and information granule, is shown in Algorithm 1.

4. Experiments

This section details the experimental datasets and results. The main goal of this section is to validate whether our proposed algorithm is effective for detecting rumor events. More specifically, Section 4.1 gives a detailed description of our experimental datasets. The construction approaches of different sentiment dictionaries are discussed in Section 4.2. And the effectiveness of the different event posts series division methods is compared in Section 4.3. Besides, the multiple sentiment encoding ways into word representations and the different types of sentimental information for rumor events detection are presented in Sections 4.4 and 4.5, respectively. The importance of encoding sentimental information into rumor events detection models is studied in Section 4.6. Section 4.7 reports the experimental results on real Chinese and English dataset, carried out by the SD-TsDTS-CGRU approach and the baseline rumor events detection models. In the end, two practical case studies are in Section 4.8.

4.1. Datasets

In our experiments, three datasets are used to make performance evaluations.

Algorithm 1: TsDTS.

```
Input: $E_i = \{(p_{i,j}, t_{i,j})\}_{j=1}^{n_i}$. $K$ is the number of intervals to be
       divided of each event. $N = \lfloor \frac{K}{2} \rfloor$ is the number of
       cluster centers. $U_i$ is the time span of event $E_i$. $l_1$ and
       $l_2$ are the pruning operators.
Output: $u_{i,k} = \{u_{i,1}, u_{i,2}, \ldots, u_{i,K}\}$ is the $i$th event posts time
        intervals.
/* Initialization */
$E_i \leftarrow E_i$ sorted by $t_{i,j}$ ;
$T_i^{\min} \leftarrow \min\{t_{ij}\}$, $T_i^{\max} \leftarrow \max\{t_{ij}\}$, $U_i \leftarrow [T_i^{\min} - l_1, T_i^{\max} + l_2]$ ;
if $n_i \geq K$ then
    $v_{i,n} \leftarrow \{v_{i,1}, v_{i,2}, \ldots, v_{i,N}\}$ cacluated by FCM, where $v_{i,n}$ is
    the $n$-th cluster center and $n \leftarrow [1, N]$ ;
    /* Calculate the middle point of two adjacent cluster centers */
    $m_{i,n} \leftarrow \frac{v_{i,n}+v_{i,n+1}}{2}$, where $n \leftarrow [1, N-1]$;
    /* Calculate the first-step division intervals */
    $T_{i,1} \leftarrow \{t_{i,j} \in T_i | T_i^{\min} \leq t_{i,j} \leq m_{i,1}\}$;
    $T_{i,N} \leftarrow \{t_{i,j} \in T_i | m_i^{N-1} \leq t_{i,j} \leq T_i^{\max}\}$;
    $T_{i,n} \leftarrow \{t_{i,j} \in T_i | m_{i,n-1} \leq t_{i,j} \leq m_{i,n}\}$, where $n \leftarrow [2, N-1]$;
    /* Calculate optimal upper and lower boundaries based on
    information granule */
    for each $T_{i,n}$ do
        update $V(a_{i,n})$ and $V(b_{i,n})$ according to Eq. 17 and Eq.
        18, respectively;
        $V(b_{i,opt}) \leftarrow \max_{b_i \geq med(T_{i,n})} V(b_{i,n})$; $V(a_{i,opt}) \leftarrow$
        $\max_{a_i \leq med(T_{i,n})} V(a_{i,n})$
    end
    /* Calculate the second-step division intervals */
    $u_{i,k} \leftarrow \{u_{i,1}, u_{i,2}, \ldots, u_{i,K}\}$ calculated by Eq. 20 or 21;
    $k \leftarrow 1$;
    while $k \leq K$ do
        $x \leftarrow$ the lower bound of $u_{i,k}$; $y \leftarrow$ the upper bound of
        $u_{i,k}$;
        $u_{i,k} \leftarrow (p_{i,x}, \ldots, p_{i,y})$;
        $k++$;
    end
else if $n_i < K$ then
    $k \leftarrow 1$ ;                    /* Calculate the intervals */
    while $k \leq n_i$ do
        $u_{i,k} = (p_{i,k})$ ;
        $k++$ ;
    end
    while $k < K$ do
        $u_{i,k} = (Null)$ ;              /* Fill in Null */
        $k++$ ;
    end
return $u_{i,k}$ ;
```

Dataset 1. A large amount of original Chinese microblogs dataset. The original Chinese microblogs are collected from Sina Weibo. The number of well-structured Chinese microblogs is about 8.11 million, which is accessible in our google drive[4]. This microblog dataset is used to,

- Train a word2vec model for Chinese microblogs.
- Collect the commonly used emoticons.
- Analyze the co-occurrence of emoticons and sentiment words.

Dataset 2. A public Chinese rumor events dataset from Sina Weibo. The dataset published by et al. [16] is crawled

[4] https://drive.google.com/open?id=1_LNC9D1baGSpaKFE6KrhsKn21x19-J8m.

Table 2
Statistics of the Chinese and English rumor events dataset.

Statistic	Sina Weibo	Twitter
Involved users	2,819,338	491,229
Total posts	3,752,459	1,101,985
Total events	4664	992
Total rumor events	2313	498
Total non-rumor events	2351	494
Average posts perevent	804	1111
Minimum posts per event	10	10
Maximum posts per event	59,318	62,827
Average time length per event	1,808.74 H	1,582.60 H
Minimum time length per event	0.02 H	0.07 H
Maximum time length per event	34,312.00 H	57,425.47 H

from the “Sina Community Management Center”[5], which is a crowdsourcing-based rumor detection platform. In this platform, all users of Sina Weibo can report the suspicious information to them. After the community administrator reviews, the audit results will be published. Unlike previous studies on Twitter where the labeling of rumors is done manually by the participants of the experiments, the official nature of this service ensures the high quality of the dataset.

The public rumor events dataset contains 2313 rumor events and 2351 non-rumor events, in which the posts related to the same events are aggregated, and each event is labeled to 1 for rumor and 0 for non-rumor. All rumor events detection models are performed on this dataset. Table 2 gives statistical details of the Sina Weibo dataset.

Dataset 3. A public English rumor events dataset from Twitter. The dataset is also published by Ma et al. [16]. The public rumor events dataset contains 498 rumor events and 494 non-rumor events, in which the posts related to the same events are aggregated, and each event is labeled to 1 for rumor and 0 for non-rumor. Table 2 gives statistical details of this public dataset.

4.2. Evaluations of sentiment dictionary construction approaches

This section mainly testifies the effect of different sentiment dictionary construction approaches proposed in this paper, including the Extended Sentiment Word Dictionary (ESWD) construction approach and the Sentiment Emoticon Dictionary (SED) construction approach. All Experiments in this section are in Chinese microblogs, and English microblogs are the same.

4.2.1. Evaluations of ESWD construction approach

We use the first dataset to train a word2vec model. Based on the relatedness of the words calculated by word2vec, we apply the proposed word-extension method to obtain the similar words of the seed sentiment words. For each sentiment category, ten seed basic sentiment words are manually selected, seventy words in total. Our word-extension method is employed to obtain extended sentiment words of the seeds from the first dataset. Fig. 3 shows that the count of the discovered extended sentiment words falls dramatically with an increase in similarity. In addition, words with similarity equal to or greater than 0.5 have relatively similar meanings in semantics and sentiment, although there are different extension counts for different sentiment categories. Thus, 0.5 is chosen as the threshold value for word similarity.

Through refining the result of word2vec, we implemented the proposed majority voting algorithm and compared its performance with the other approach, the original word2vec tool. Table 3 shows the extended sentiment words of the two seed basic sentiment words, “Rejoice (欢喜)” and “Despise (鄙视)”, discovered by the two approaches. The table notes that the real extended sentiment words are found manually, denoted by the bold font. From the experimental results, we can observe that the original word2vec tool discovers a lot of related words of the seed sentiment words, but there are also some noise words. For example, the seed word“Despise (鄙视)” is totally different from “Pity (同情)”, an antonym. As we stated, our approach can obtain more accurate extensions than word2vec due to the proposed noise filtering method using the triangle-relations-based majority voting algorithm. As result, given 70 seed basic sentiment words in 7 sentiment categories, the original word2vec tool obtained 11,076 candidate extensions with an accuracy of 59.82%. Our approach, in contrast, yielded 4226 extensions with an accuracy of 76.88%. What’s more, our approach also can be easily adapted to other domains, so long as it is trained using text data in the corresponding domains.

In addition, to testify the identification methods of sentiment orientations and sentiment categories for ESWD, this paper introduces the sentiment concordance between extended sentiment words and basic sentiment words. We observe that parts of extended sentiment words will appear in the original BSWD and the above identification methods are also employed in these overlapping words. That is the accuracy of the proposed method can be evaluated by comparing the consistency of our methods and human annotation. As result, given 70 seed basic sentiment words in 7 sentiment categories, we obtained 4226 extensions, of which 690 words are in the BSWD. That is to say, the accuracy of sentiment orientations identification of ESWD is 95.07% and the accuracy of sentiment category classification of ESWD is 86.23%. Besides, some examples of ESWD are shown in Table 4 and the complete ESWD dataset is accessible in our google drive[6].

4.2.2. Evaluations of SED construction approach

To provide a deep investigation of our sentiment orientations and sentiment categories identification methods for emoticons, two annotators are asked to annotate each emoticon following two steps. First, determine if a given emoticon is positive or negative, and assign 1 to positive ones and 0 to negative ones. Then, label each emoticon with one of the seven sentiment categories. Two annotators independently annotate each emoticon and discuss to reach an agreement. The inter-annotator agreement is 95% for sentiment orientations and 70% for categories. We can observe that there is a large bias in the perception of sentiment information between different individuals, particularly for sentiment category. It also indicates that the necessity to design a dynamic approach to identify the complicated and changeable sentiment information of emoticons. In this paper, we combine the two annotations to obtain the standard manual annotation, which is accessible in our google drive[7].

Besides, the accuracy of our identification methods mentioned-above in Section 3.3.3 is 94.5% for sentiment orientations and 79.5% for sentiment categories, which have reached, even exceeded, the manual annotation. What’s more, our approach can automatically update the sentiment information of emoticons on a regular basis, which not only reduces the work of human beings in annotations, but also can capture the sentiment changes dynamically during the usage of emoticons.

4.3. Evaluations of the event posts series division methods

As mentioned above, the time series division methods for rumor events detection can be divided into two parts: time-equal-length based time series division (TETS) [16,17] and posts-equal-length based time series division (PETS) [18]. In addition, to testify

[5] http://service.account.weibo.com/?type=5&status=0.

[6] ESWD:https://drive.google.com/file/d/1rG3ulAbzXuQdx09GUD2kJNAO6vZXL9fQ.

[7] SED: https://drive.google.com/file/d/1VKNX92by5gQfP2Dmab51gl-49eOk8O6y.

Table 3
Illustration of performance comparisons for two word-extension approaches.

Seed #1	Rejoice/欢喜
Word2vec	**Delight/欣喜, Jubilate/欢欣, Joy/喜乐, Joyful/喜悦**, Free/自在, Melancholy/惆怅, **Happy/开心, Gambol/雀跃**, Anxious/忧, **Jubilant/开怀**, Worry/牵挂, Concerned/挂念, Reluctant/不舍, **Delightful/欢愉**, Bittersweet/苦乐, Sorrowful/感伤, **Pleased/舒心**, Glorious/美好, Sperturbed/忐忑, Indifferent/淡然, Homesick/思乡, Content is happiness/知足常乐, **Peaceful and happy/安乐, Happy/快乐 …**
Our Algorithm	**Delight/欣喜, Jubilate/欢欣, Joy/喜乐, Joyful/喜悦**, Free/自在, **Happy/开心, Gambol/雀跃, Jubilant/开怀**, Worry/牵挂, **Delightful/欢愉**, Homesick/思乡, **Peaceful and happy/安乐, Happy/快乐**
Seed #2	Despise/鄙视
Word2vec	**BS, Condemn/谴责, Spurn/唾弃, Detest/痛恨**, Denounce/声讨, Pity/同情, Resist/抵制, Deceive/愚弄, **Indignation/愤慨, Hate/讨厌**, Insult/侮辱, **Rebuke/斥责, Mai Tai/埋汰, Dislike/不待见**, Cheat/蒙骗, Unfollow/取关, **Revile/唾骂, NC, Fuck, Be unwilling to mention/不齿**, Flatter/吹捧, Agree/同意, **Wordless/无语 …**
Our Algorithm	**BS, Condemn/谴责, Spurn/唾弃, Detest/痛恨**, Denounce/声讨, Resist/抵制, **Indignation/愤慨, Hate/讨厌**, Insult/侮辱, **Rebuke/斥责, Mai Tai/埋汰, Dislike/不待见**, Unfollow/取关, **Revile/唾骂, NC, Fuck, Be unwilling to mention/不齿, Wordless/无语**

Table 4
Illustration of performance comparisons for two word-extension approaches.

Extended sentiment word	Sentiment orientation	Sentiment category	Basic sentiment word
Kai-Sen/开森	1	Happiness/乐	Happy/开心, Joyous/欢乐
Giggling/咯咯	1	Happiness/乐	Laughing and joking/嘻嘻哈哈, Laughingly/笑哈哈
Pray/祈望	1	Like/好	Bless/保佑
Yun-Cai/晕菜	−1	Surprise/惊	Astonished/惊奇
Person Pork/人肉	−1	Disgust/恶	Damn/该死, Curse/诅咒
Injury body/伤身	−1	Sadness/哀	Nerve-racking/伤神
Low ebb/低谷	−1	Sadness/哀	Low tide/低潮

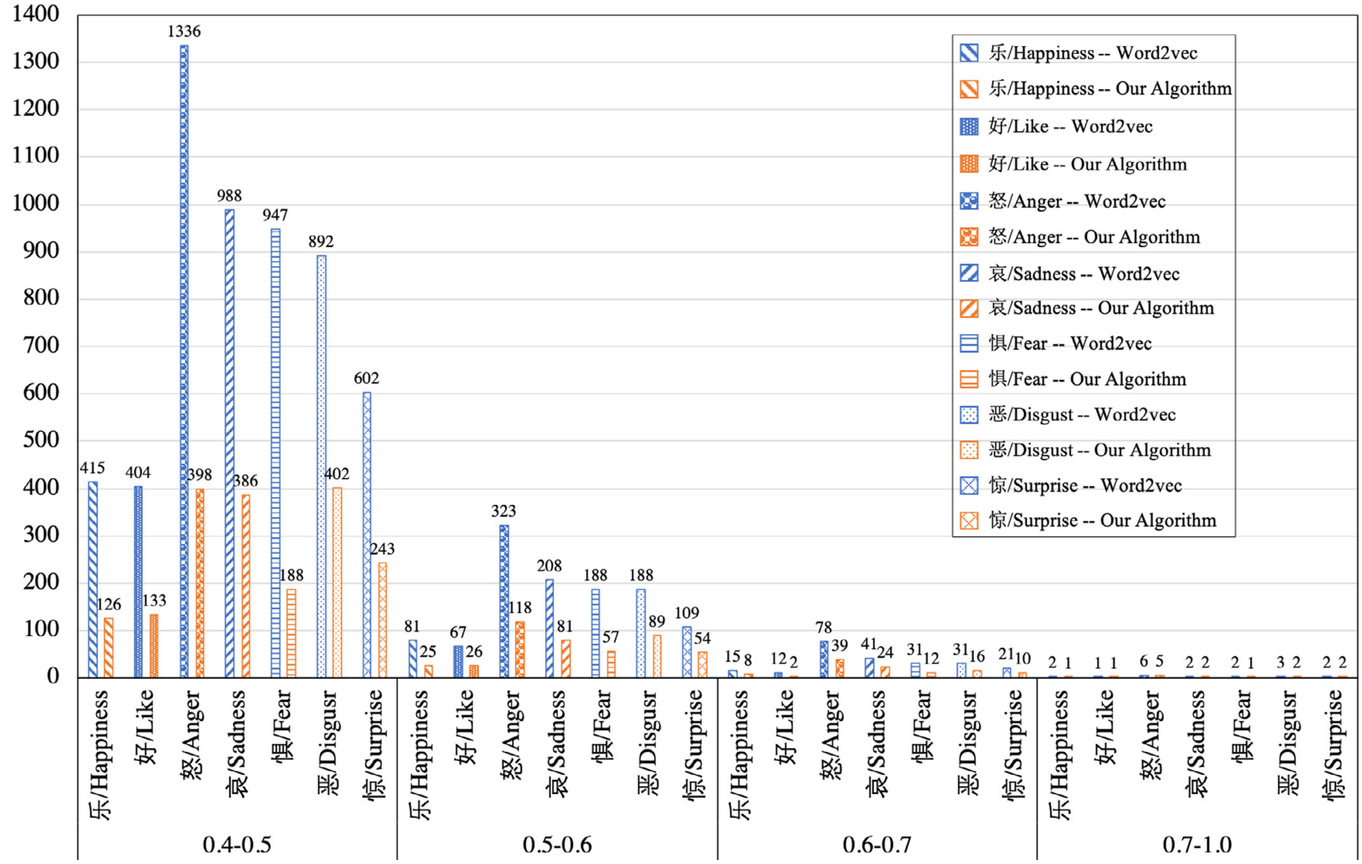


Fig. 3. The relationship between word count and word similarity.

the superiority of the TsDTS which encodes the sentimental information into the division process, the two other division methods are also be compared in this paper: (1) One-step Dynamic Time Series (OsDTS) division algorithm, which divides the posts series only based on the fuzzy clustering algorithm. (2) TsDTS Posts-only, which only takes the number of posts into consideration in the second step of TsDTS. Two sets of experiments are conducted in this paper to evaluate the different event posts series division methods, including *comparison of division time*, and *detection performance comparison based on different division methods*. For a fair comparison between the six time-series division methods, each experiment is under the same experimental environment and

Table 5
The time consumption of different division methods.

	TETS [16]	TETS [17]	PETS [18]	OsDTS	TsDTS Posts-only	TsDTS
Thread-1	20.142	19.618	17.625	2629.230	595.729	630.896
Thread-2	19.159	18.764	16.756	2727.919	560.808	595.371
Thread-3	18.440	17.850	15.988	2400.751	529.934	563.467
Thread-4	18.264	17.240	15.579	2359.300	512.351	540.632
Thread-5	18.239	17.496	15.494	2381.879	520.154	546.696
Thread-6	18.992	18.252	16.340	2404.590	544.976	578.507
Thread-7	19.428	18.633	16.700	2460.932	560.421	596.950
Thread-8	16.525	15.639	13.575	2167.722	464.597	485.839
Thread-9	20.403	19.868	17.896	2584.680	608.292	646.344
Thread-10	20.860	20.347	18.377	2669.068	633.154	683.991

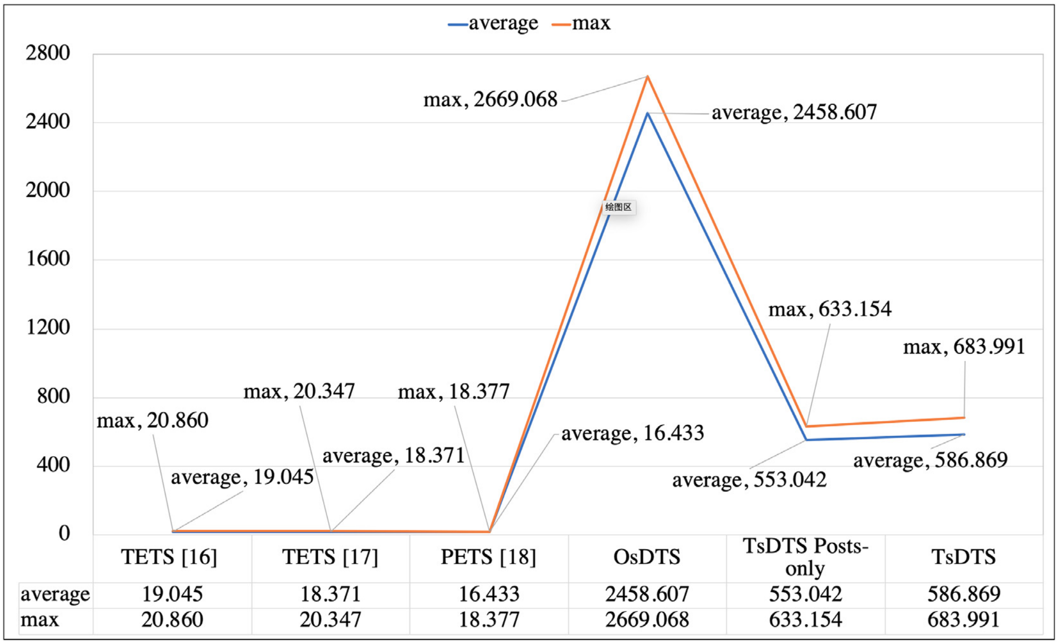


Fig. 4. The maximum and average time consumption of each method.

the same two-layer GRU model. And this model outperforms other neural networks in rumor events detection, such as tanh-RNN, LSTM, etc [16].

4.3.1. *Experiment one: comparison of division time*

In the first experiment, ten threads (can be more) are used to accelerate the independent division process of each method on the real **Dataset 2**. Table 5 shows the time consumption of each method on each thread (in seconds) and Fig. 4 show the maximum and average time consumption of each method. Besides, since the ten threads are parallel, the maximum time consumption exactly is the total time spent of each method in the experiment.

Based on the data in Table 5, whether it is the time-equal-length or post-equal-length division methods, the whole process of these three average division methods approximately takes anything from 15 s to 20 s. Among them, the average and maximum time-consumption of TETS [16] is slightly higher than the two others (like Fig. 4, average: 19.045 vs. (18.371, 16.433); max: 20.860 vs. (20.347, 18.377)). It is chiefly because the TETS [16] needs to search the global largest continuous interval by iterating, which is more time consuming than the direct average division methods of TETS [17] and PETS [18].

Furthermore, the dynamic division methods obviously take longer than the average ones, as shown in Table 5 and Fig. 4. The average time of each thread of the OsDTS method is more 2400 s and the maximum time is more than 2600 s. The other two dynamic division methods also reach about 550 s and 650 s on average and maximum, respectively. Mainly because the dynamic division methods need extra time to cluster for all the posts of each event, relative to the direct average division methods. In the experimental dataset (**Dataset 2**), the total number of posts in the event reaches 3.75 million. At the same time, the average number of posts per event also exceeds 800 and the most one is about 59,318. Thus, the division time increase is interpretable and reasonable in dynamic division methods.

4.3.2. *Experiment two: detection performance comparison based on different division methods*

In the second experiment, we evaluate the performance of the rumor events detection algorithm based on these six different division methods. Table 6 shows the detection results achieved by these algorithms on accuracy (Acc.), precision (P), recall (R) and F-Measure (F). Meanwhile, Fig. 5 and Fig. 6 are the precision-recall curve and ROC curve, respectively.

As shown in Table 6, the performance of TETS [16,17] is worse than PETS [18] (about 1.5% to 3.6% in Acc. and 0.0% to 4.8% in F). From Table 2 we can see that the maximum posts per event are 59,318 and the posts are not evenly distributed in the time dimension. Therefore, the fixed time length will cause the excessive number of microblog posts in part of the time series and a few

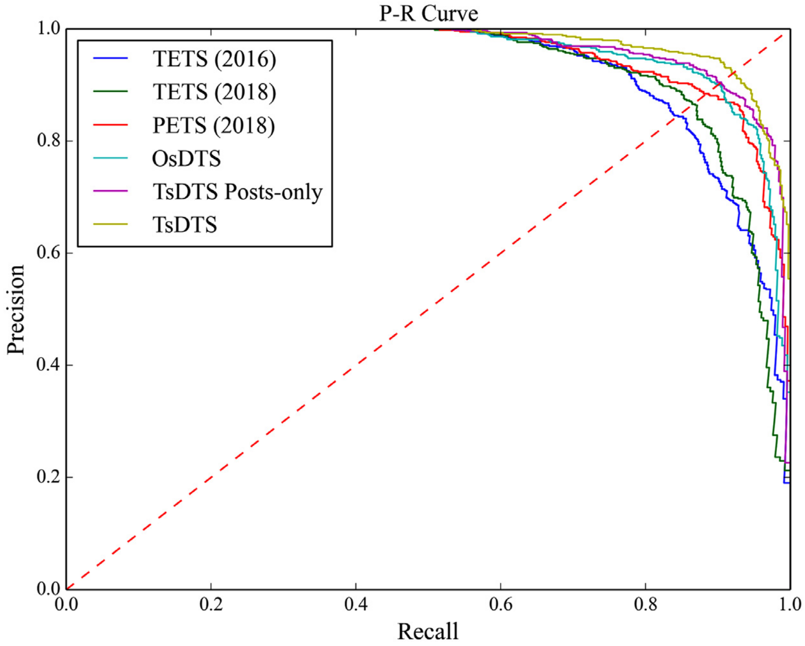


Fig. 5. The precision-recall curve of these six algorithms.

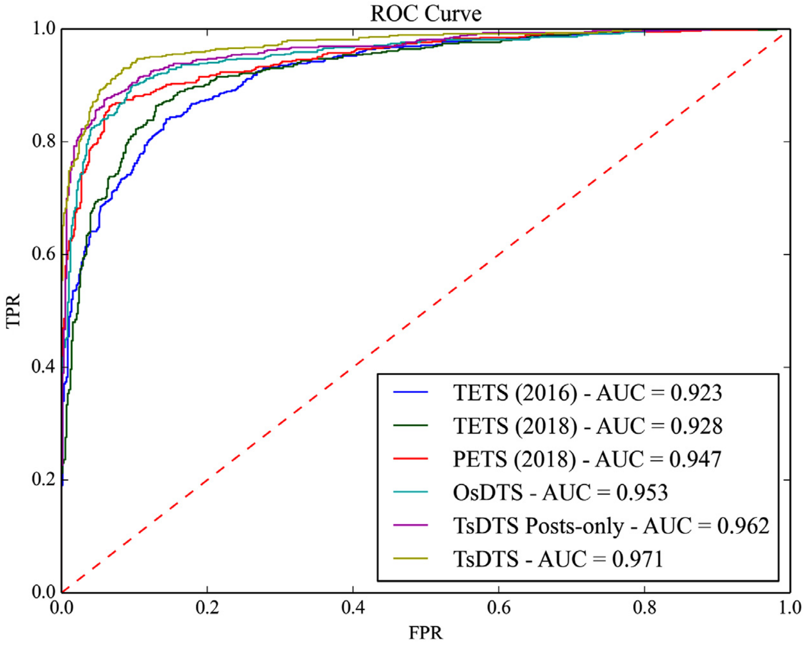


Fig. 6. The ROC curve of these six algorithms.

Table 6
The detection performance based on different time series division methods.

TS-Method	Class	Acc.	P	R	F
TETS (2016) [16]	R	0.840	0.824	0.869	0.846
	NR		0.858	0.811	0.835
TETS (2018) [17]	R	0.861	0.840	0.895	0.867
	NR		0.885	0.827	0.855
PETS (2018) [18]	R	0.876	**0.938**	0.806	0.867
	NR		0.828	**0.946**	0.883
OsDTS	R	0.899	0.905	0.895	0.900
	NR		0.894	0.905	0.899
TsDTS Posts-only	R	0.905	0.925	0.883	0.903
	NR		0.886	0.927	0.906
TsDTS	R	**0.920**	0.926	**0.915**	**0.920**
	NR		**0.915**	0.926	**0.920**

numbers of posts in some others. On the contrary, the posts-fixed based method guarantees the reasonable number of posts in each time series and achieves higher performance than the time-fixed based method. However, these two types of methods are weak and ineffective in accuracy compared with the dynamic division methods. What's more, the best dynamic division method TsDTS is accurate at 92.0% (Table 6), while the AUC value also reaches 0.971 (Fig. 6). In addition, according to the balance point of Fig. 5 and the AUC value of Fig. 6, TsDTS Posts-only is pretty much with OsDTS. However, the division time consumption of TsDTS Posts-only is only one-fifth of OsDTS's, seen from Fig. 4. But all in all, both the two two-step dynamic division methods outperform the one-step dynamic division method. There is because our dynamic division

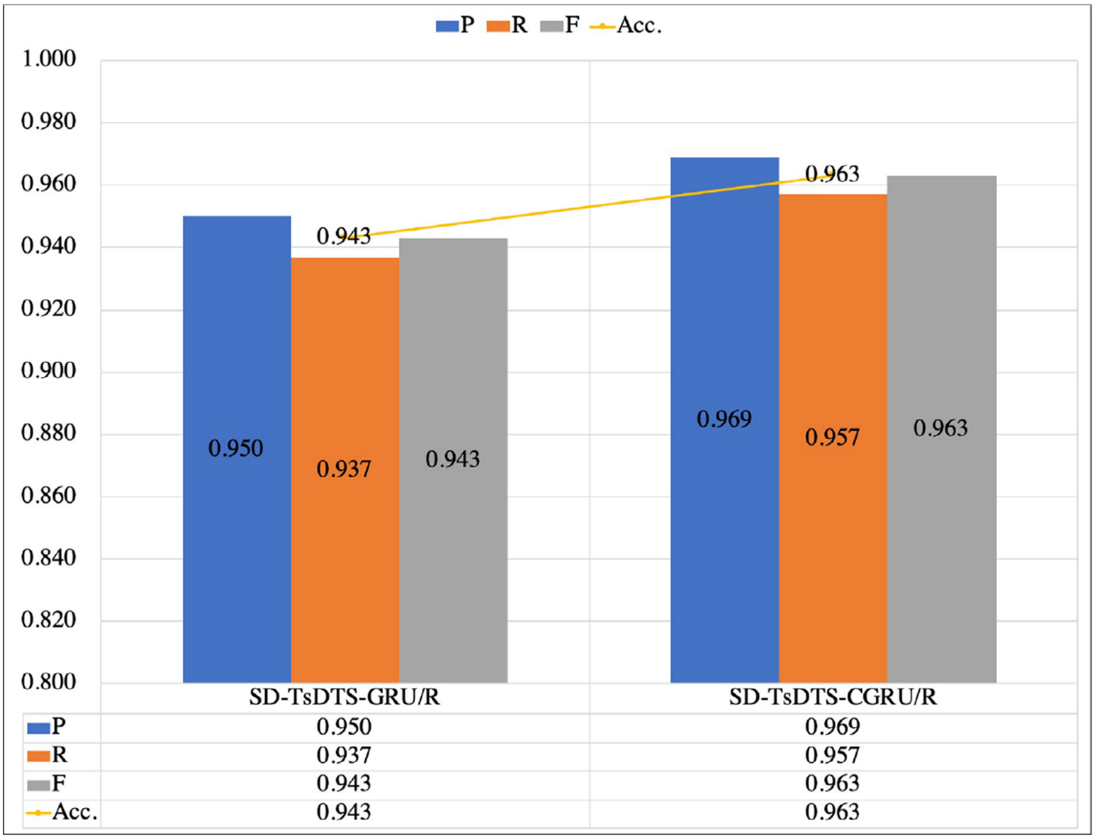


Fig. 7. Detection performance of different ways to encode sentiments into word representations (Rumor class).

methods consider the distribution characteristics of event posts, which are useful in rumor events detection.

4.4. Evaluations of multiple sentiment encoding ways into word representation

In this section, two experiments are conducted to testify the proposed two-layer cascade GRU model can more effectively capture the variation of contextual and sentimental information of each event over time. For a fair comparison, each experiment is under the same experimental environment, including experimental dataset (**Dataset 2**), dynamic division method (TsDTS), sentimental information (all in SD) and model parameters. Especially, sentiment words and emoticons are both grouped by Sentiment Categories and fed into the input layer in these two experiments. Using this kind of grouping is better than grouping by Sentiment Orientations, proved in Section 4.5.

The first experiment, SD-TsDTS-GRU, concatenates word embedding with sentiment embedding directly, then encodes them uniformly by a two-layer GRU model. SD-TsDTS-CGRU, the other experiment, uses a two-layer GRU model to encode word embedding and sentiment embedding separately, which are concatenated next for the subsequent detection process. Fig. 7 shows the detection performance of each experiment on accuracy (Acc.), precision (P), recall (R) and F-Measure (F).

Fig. 7 is a good illustration that the way of encoding word embedding and sentiment embedding separately can obtain a better detection performance than directly concatenated, where Acc. and F both improve 2%. To our knowledge, it is perfectly normal that sentimental information is sparse in the posts of some events. If sentiment embedding and word embedding are concatenated directly, the sparsed sentimental information will be covered by the rich words' information. Consequently, it will probably be difficult for the detection model to capture the variation of sentimental information over time for rumor events detection. Instead, the separate sentimental information encoding can preserve the variation of sentimental information over time in a better manner, while achieving a higher dimensional sentimental feature representation.

4.5. Effects of different sentimental information for rumor events detection

This section is mainly to testify the enhancement of sentimental information and compare the effects of different types of sentimental information for rumor events detection. The basic model is a two-layer CGRU model based on TsDTS, whose input of each interval is the accumulation of all the word embedding in the current interval. Moreover, the other models extend their inputs by grouping different entries in the sentiment dictionary according to their sentiment information. In which sentiment dictionary includes Sentiment Word Dictionary (SWD) and Sentiment Emoticon Dictionary (SED), and sentiment information involves Sentiment Category (SC) and Sentiment Orientation (SO). There are seven models designed to study the above goals, as listed in Table 7.

Fig. 8 shows the detection performance of all experiments on the Sina Weibo dataset from the Rumor Class aspect. The results indicate that, overall, the accuracy of sentiments enhanced approaches is generally higher than the TsDTS-CGRU model without any sentimental information, about 0.2–4.3%. It further proves the effectiveness of sentimental information for rumor events detection.

For the methods of grouping sentiment entries by sentiment orientations, the sentiment orientations of SED (TsDTS-CGRU + SED_SO) is more effective in rumor events detection than the

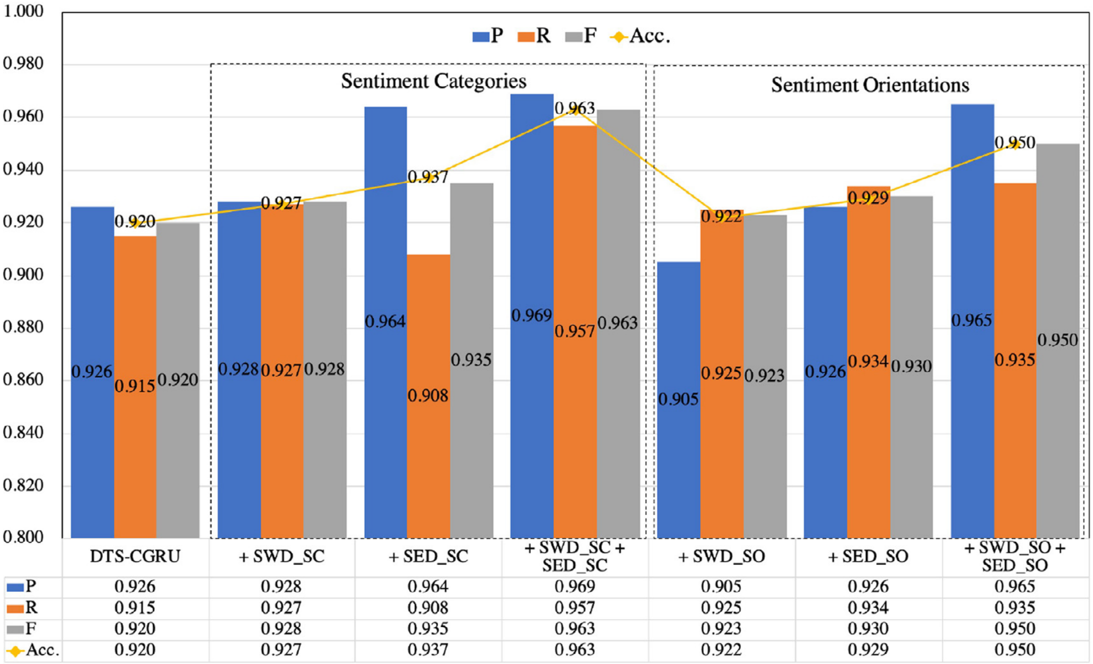


Fig. 8. Performance of different sentimental information combinations (Rumor class).

Table 7
Different types of sentimental information for rumor events detection.

Basic model	Encode sentiments	SC		SO	
		SWD	SED	SWD	SED
TsDTS-CGRU	–	×	×	×	×
	+ SWD_SC	✓	×	×	×
	+ SED_SC	×	✓	×	×
	+ SWD_SO	×	×	✓	×
	+ SED_SO	×	×	×	✓
	+ SWD_SC + SED_SC	✓	✓	×	×
	+ SWD_SO + SED_SO	×	×	✓	✓

SWD's (TsDTS-CGRU + SWD_SO) (Acc. 92.9% vs. 92.2%). The detection performance of the methods of grouping sentiment entries by sentiment categories is in the same situation. Part of this is due to the number of extended sentiment words is more than emoticons, which imports noise and decreases the accuracy greatly. On the other hand, the text is more of a factual description of an event, while emoticons are more focused on the expression of individuals' emotions in OSNs, such as Happiness, Anger, Sadness, etc. What's more, the performance of the methods of grouping sentiment entries by sentiment orientations is worse than the ones by sentiment categories, and the difference is about 0.5–4.1% in accuracy. Probably because the sentiment orientations (positive or negative) cannot adequately describe the connotation of words or emoticons. At the same time, the subdivision of sentiments further explores sentimental information of words and emoticons. Of course, the integration of emoticons and sentiment words also gives a greater performance increase for rumor events detection. Especially, the method embedding sentiment categories of emoticons and sentiment words (TsDTS-CGRU + SWD_SC + SED_SC) achieves top accuracy about 96.3%, which is selected as our rumor events detection method in this paper, named as SD-TsDTS-CGRU.

4.6. Study the importance of encoding sentimental information into rumor events detection models

To verify the effectiveness of encoding sentimental information into time series division and word representations for rumor events detection, the following four models experiment in this paper and the experimental results are listed in Table 8.

(1) **Model-1.** Sentimental information is not encoded into time series division and word representations. That is, TsDTS Post-only, proposed in Section 4.3.2, is adopted for the time series division in this model. Same as [16], only the two-layer GRU model is used to encode the word embedding.
(2) **Model-2.** In this model, sentimental information is only encoded into time series division with the Two-steps Dynamic Time Series (TsDTS) algorithm.
(3) **Model-3.** Sentimental information is only encoded into word representations with a two-layer GRU model. And the time series division is the same as the **Model-1**.
(4) **Model-4.** It is the SD-TsDTS-CRGU proposed in this paper, which encodes sentimental information into time series division and word representations.

From the above table, against previous detection studies, **Model-1** achieves competitive performance, which only adopts the preliminary time series division algorithm and word embeddings. The introduction of sentimental information into time series division in **Model-2** gains at least 1.5% accuracy improvement to **Model-1**, mainly because human emotional reactions to each event reflect the development of the event to a certain extent. When sentimental information is encoded into time series division, the division of time series can reflect the development trend of each event in a better manner, thereby assisting rumor events identification. In **Model-3**, the detection performance gets a promotion of 4.5% in accuracy to **Model-1**, even achieves the best precision of 97.4% (R) among all models. Encoding the sentimental information into word representations can express users' emotional change to different events in a better manner. It also illustrates the impor-

Table 8
Performance of encoding sentimental information into different detection processes.

Model	Time series division	Word representations	Class	Acc.	P	R	F
Model-1	×	×	NR	0.905	0.925	0.883	0.903
			NR		0.886	0.927	0.906
Model-2	✓	×	NR	0.920	0.926	0.915	0.920
			NR		0.915	0.927	0.920
Model-3	×	✓	NR	0.946	**0.974**	0.917	0.945
			NR		0.920	**0.976**	0.947
Model-4	✓	✓	NR	0.963	0.969	**0.957**	**0.963**
(ours)			NR		**0.957**	0.969	**0.963**

tance of sentimental information for rumor events detection. Obviously, the encoding of sentimental information into time series division and word representations respectively can apparently improve the accuracy of rumor events detection, and the performance is the best (Seen the **Model-4** in Table 8).

4.7. Evaluations of rumor events detection models

To verify the effectiveness and superiority of our approach, SD-TsDTS-CGRU algorithm is evaluated with other five rumor events detection algorithms, including the machine learning methods of DT-Rank (2015), SVM-TS (2015), LK-RBF (2016), SVM_{all}^{DTS}(2018) and the deep learning methods of GRU-2 (2016), CallAtRumors (2018). The experimental results on the two-real dataset (Sina Weibo/**Dataset 2** and Twitter/**Dataset 3**) are discussed in Section 4.7.1 and Section 4.7.2, respectively.

DT-Rank [61]: A decision-tree-based ranking model can recognize treading rumors, relies on inquiry phrases and statistical features. We make it comparable to our model by realizing their features.

SVM-TS [5]: Ma et al. proposed an SVM classifier, based on time-series structures, to capture the variation of social context features. These content-based, users-based, and propagation-based features are also implemented in our paper to compare with our SD-TsDTS-CGRU method.

LK-RBF [62]: In this study, the author proposed two methods to address the problem of implicit data, which is based on hashtags and web links into a conversation. Only the link-based method with the Radial Basic Function (RBF), achieved the best performance in their experiment, is selected in our paper.

SVM_{all}^{DTS} [11]: Wang et al. introduced three new features based on the communication theory of rumor events in sociology and constructed a model to capture the dynamic time series features over time, which are both implemented in this paper.

GRU-2 [16]: RNN is introduced for learning the hidden representations for rumor detection. They realized three types of RNN-based algorithms, including tanh-RNN, LSTM-1/LSTM-2 and GRU-1/GRU-2. GRU-2 achieved the best performance by capturing higher-level feature interactions, which is realized in this paper.

CallAtRumors [18]: Chen et al. presented a deep attention based RNN model to learn hidden representations of posts-equal-length time series for detecting rumor events. Following the settings in their work, we also choose this method as baseline in this paper.

We implement SVM models using LibSVM and RNN models with Keras. We split the events with a ratio of 3:1 for training and test respectively. Besides, Accuracy (Acc.), Precision (P), Recall (R) and F-Measure (F) are selected as evaluation metrics. N is empirically set to 50, which is same with [16].

4.7.1. Evaluation on real chinese dataset - Sina weibo/***dataset 2***

Table 9 shows the performance of all the systems on Chinese Sina Weibo. Our method outperforms all the baselines on Sina Weibo dataset. DT-Rank cannot effectively identify the rumor events with the set of regular expressions, which is contained by only 1.63% Sina Weibo posts. That is why the results of DT-Rank are not satisfactory. LK-RBF and SVM-TS achieve better results with more features extracted from microblog posts, indicating the ability of feature engineering to help classifiers detect rumors better. Especially, compared with the DT-Rank model, the accuracy of the SVM-TS model is increased by 14.8% and the F-measure is increased by 13.6–16.1%. Part of this is due to the good generalization ability of SVM model itself. The most important is that the SVM-TS model takes the features variable over time into consideration. In addition, SVM_{all}^{DTS} outperforms other machine learning based baselines because of the temporal structures SVM_{all}^{DTS} involved, but they are all still worse than deep learning models.

Table 9
Performance of rumor events detection methods on Sina Weibo.

Method	Class	Acc.	P	R	F
DT-Rank (2015) [61]	R	0.648	0.683	0.624	0.652
	NR		0.615	0.674	0.643
LK-RBF (2016) [62]	R	0.681	0.768	0.629	0.692
	NR		0.604	0.749	0.559
SVM-TS (2015) [5]	R	0.796	0.813	0.763	0.788
	NR		0.780	0.828	0.804
SVM_{all}^{DTS} (2019) [11]	R	0.849	0.868	0.820	0.843
	NR		0.832	0.878	0.854
GRU-2 (2016) [16]	R	0.833	0.829	0.842	0.835
	NR		0.837	0.825	0.830
CallAtRumors (2018) [18]	R	0.866	0.872	0.867	0.869
	NR		0.859	0.865	0.862
SD-TsDTS-CGRU	R	**0.963**	**0.969**	**0.957**	**0.963**
	NR		**0.957**	**0.969**	**0.963**

For the three deep-learning based methods, CallAtRumors achieves higher performance than GRU-2 (about 3.3% in accuracy and 3.2% in F-measure) by involving the temporal long-term characteristic and deduping the same textual phrases. On the Sina Weibo dataset, SD-TsDTS-CGRU outperforms competitors by achieving the accuracy, precision, recall and F-measure of 0.963, 0.969 (R), 0.957 (R) and 0.963 (R) respectively. The effectiveness validation proves the reasonable time series method can retain the distribution characteristic of rumor and non-rumor events and the event sentiments can enhance the performance of detecting rumor events.

4.7.2. Evaluation on real english dataset - Twitter/***dataset 3***

The detection performance of all algorithms on Twitter (**Dataset 3**) is listed in Table 10. Overall, all models perform less on the Twitter dataset than the Sina Weibo dataset. The analysis found that, on one hand, the number of events in the Twitter dataset is only one-fourth of the Sina Weibo dataset, which limits the effect of the detection model. On the other hand, the Twitter dataset contains some texts in other languages, such as Japanese, Korean, Arabic, etc. They are all deleted during processing, which may result in partial loss of information. But all in all, like the detection performance on the Chinese dataset, the method proposed

Table 10
Performance of rumor events detection methods on Twitter.

Method	Class	Acc.	P	R	F
DT-Rank (2015) [61]	R	0.644	0.638	0.675	0.656
	NR		0.652	0.613	0.632
LK-RBF (2016) [62]	R	0.675	0.785	0.605	0.684
	NR		0.591	0.775	0.671
SVM-TS (2015) [5]	R	0.774	0.763	0.779	0.671
	NR		0.771	0.782	0.776
SVM_{all}^{DTS} (2019) [11]	R	0.818	0.848	0.790	0.818
	NR		0.804	0.836	0.820
GRU-2 (2016) [16]	R	0.800	0.798	0.811	0.804
	NR		0.805	0.787	0.796
CallAtRumors (2018) [18]	R	0.833	0.826	0.837	0.831
	NR		0.816	0.854	0.835
SD-TsDTS-CGRU	R	**0.885**	**0.878**	**0.900**	**0.889**
	NR		**0.891**	**0.869**	**0.880**

Table 11
Performance of rumor events detection methods on Twitter.

Case-1	Title	Some people were poisoned by eating lobsters in Guangzhou Renhe food street last night.
	URL	https://service.account.weibo.com/show?rid=K1CaS6ABh7Kgl
Case-1	Title	Yongsheng Wang said Peking University was formerly known as Yanjing University.
	URL	https://service.account.weibo.com/show?rid=K1CaS6AFe7KYj

in this paper achieves the best detection performance. And the accuracy increases by about 5% than the latest CallAtRumors method.

4.8. Case studies

With the rapid development of OSNs, the bar for entry to information production and dissemination has been lowered, which caused hard to determine the authenticity of the information. As the largest social media platform in China, Sina Weibo has amazing amount of information every day, that's truly producing and spreading on the platform. To ensure the reliability of the platform information, Sina Weibo provides a whistleblower platform based on crowdsourcing, named "Sina Community Management Center". All Weibo users can report suspected information to the center and the result will be published after review by the community administrator. At present, the number of judged rumor events is only about 38,170 until 2019/10/12, and the average decision period is more than 24h. Hence, it is difficult for the platform to play a role in rumor events detection.

Fortunately, the automatic rumor events detection framework proposed in this paper can assist OSNs to identify false information and reduce the labor workload and judgment cycle. Two real cases on September 16th and 17th, 2019 are listed in this paper, as shown in Table 11. The details of these two events are accessible in our google drive[8]. Figs. 9 and 10 is the change of detection accuracy of the two cases as the time of event changes, respectively. At the same time, we use the official report time of rumor events as a reference, i.e., the report time over the rumor events given by the debunking services of Sina community management center.

4.8.1. Case-1

The first event occurred at 8:31 pm on September 16, 2019. A group of people was poisoned by eating lobsters in Renhe food street, Guangzhou. Since the related posts in Case-1 are less, the abscissa of Fig. 9 is in seconds.

Table 12
Parts of contents of Case-1.

Time	Content
2019/9/16 21:33	Chinese: **辟谣** !网传人和美食街食物中毒死亡事件为**不实信息**。 English: **Rumors**! The death of food poisoning in Renhe food street is **false information**.
2019/9/16 21:52	Chinese: 这个是**假新闻**。 English: This is **fake news**.
2019/9/16 21:58	Chinese: 官方已经**辟谣**了是**喝酒喝多了**。 English: Officials have **dismissed the rumor**, and they are **only drinking too much**.
2019/9/17 0:07	Chinese: **对滴** ,就是**造谣**哦。 English: **Righ**, it's a **rumor**.

Plotted on a graph, in the early stage of this event, the accuracy of rumor events detection is almost zero. This is mainly because everyone is not aware of the specific circumstances of the incident in the beginning, and is all in a state of discussion and suspicion. It is difficult for the model to detect the authenticity of this event from these texts. But at 21:33, the model has a probability of about 30% that this event is a rumor. According to analyze the posts, it is discovered that a person published a post at this time (seen the first line of Table 12), which indicated the event is not true. However, there is no more discussion at this time, hence the model cannot predict it with higher accuracy. Another important point in time, at 21:52, another user published a similar post (seen the second line of Table 12), which again pointed out the event is fake news. At the same time, the model also gives a more accurate prediction, a probability of about 84.2%. After that, more and more users found this event was a rumor (seen the third and fourth line of Table 12). Our detection model also has a probability of more than 99% that the event is a rumor. However, the official report time of Sina Weibo is at 23:03 on September 16th (seen the black vertical dotted line in Fig. 9). If 0.5 is used as the criterion for the detection accuracy of the model, our model is about 71 min earlier the official time. What's more, the model's prediction accuracy reaches over 99% about 66 min before the official time. Overall, even the event has a small number of posts, our model can still predict the authenticity of this event faster and more accurately.

4.8.2. Case-2

The second event occurred at 15:35 on September 17, 2019. Someone posted that Peking University was formerly known as Yanjing University, which was also accompanied by an old photo of the Yanjing University's gate. Case-2 has a large amount of discussed posts, thus the model detects the authenticity of the event by hours, as shown in Fig. 10. As in Case-1, the detection accuracy of our model is lower than 10% or even is zero in the first hour or two.

Yet as time passed, some users began to bemoan the poster's historical illiteracy. For example, the posts in the third hour (seen the first two lines of Table 13) and the fourth hour (seen the third line of Table 13). What's more, there are some sentiment words and sentiment emoticons contained in these posts, such as hilarious, fools, 😂 🤭 etc. At the same time, the detection accuracy of our model also starts to rise, reaches to 93.4% in the fourth hour. It is also indicated that sentimental information is useful to detect rumor events in OSNs. In addition, it should be specially stated that the detection accuracy plummets by about 10% in the 5th hour. By analyzing the posts in the 5th hour, it is found that there are some event-irrelevant discussions and disputes with a few sentimental information between several users, which have a certain impact on our model. But in general, most of the discussions between users are still more targeted to this event, and the detection accuracy is gradually increasing in the following 7th hour, 25th

[8] https://drive.google.com/open?id=1sNLe-r55m5kwl_X_RlYyU25uH4K_jEf-.

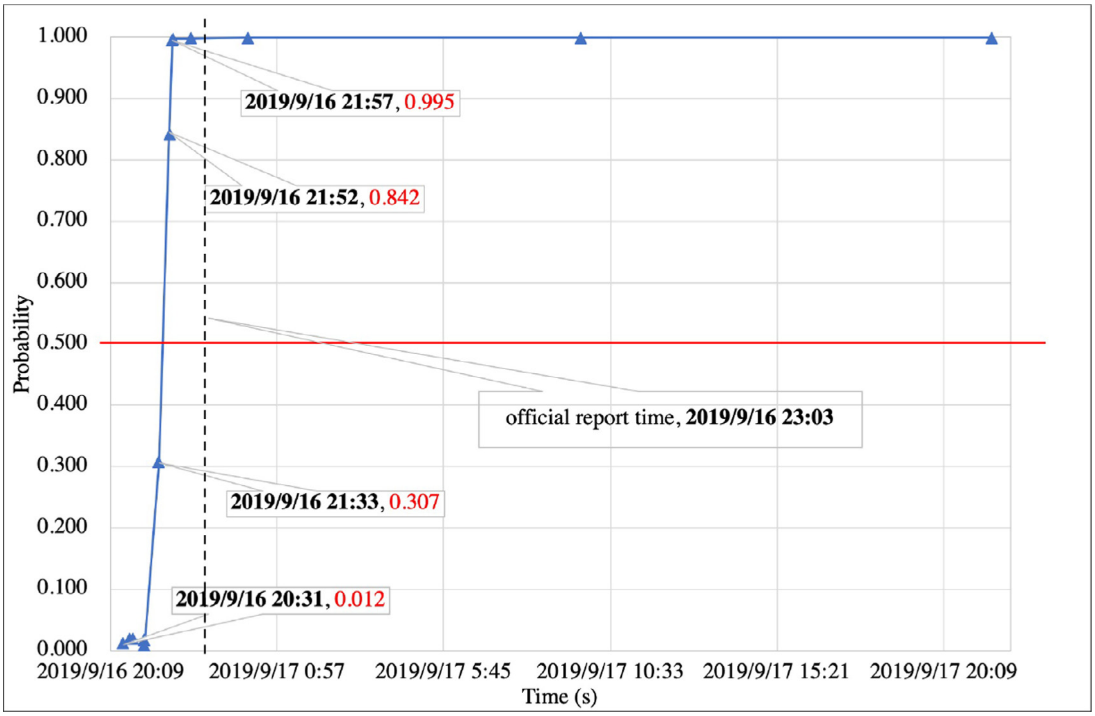


Fig. 9. Case-1: Change in probability of rumor event detection in seconds.

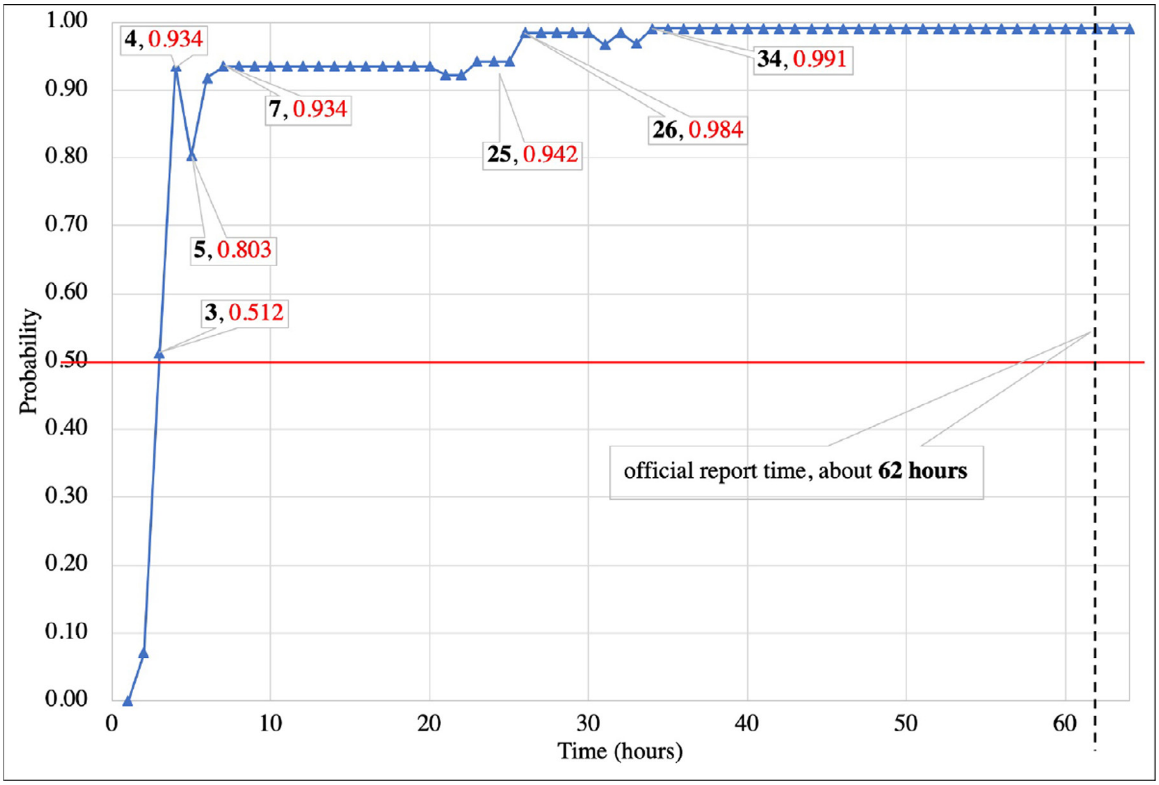


Fig. 10. Case-2: Change in probability of rumor event detection in hours.

Table 13
Parts of contents of Case-2.

Time		Content
3rd hour	2019/9/17 17:06	Chinese: 第一次听说北大前身是燕京大学这是要**笑死人** 😂😂😂😂😂 English: This is the first time I've heard that Peking University was formerly known as Yanjing University, and that will be **hilarious**. 😂😂😂😂😂
	2019/9/17 17:43	Chinese: **哈哈哈** ,一大把**哪啥**🤭 ,**真好**。 English: **Hahaha**, a lot of **fools** 🤭, very **laughable**.
4th hour	2019/9/17 18:28	Chinese: 应该是北大和燕京大学**合并**了吧 😓😓 English: The Peking University was **merged** with Yanjing University. 😓😓
5th hour	2019/9/17 19:07	Chinese: 就你这样还大学生?**哟呸**...**恶心** 🤮🤮🤮🤮🤮 English: Be like you person like this there is also a college student? **Bah**! ... **Disgusting**. 🤮🤮🤮🤮🤮
	2019/9/17 19:10	Chinese: 大学生就是对付你这种**混蛋**的。 English: College students deal with your **bastard**.
7th hour	2019/9/17 21:34	Chinese: 北大成立比燕大还**早**。 English: Peking University was established **earlier** than Yanjing University.

hour, etc. In the same vein, the model proposed in this paper is about 59 hours faster than the official report time, takes 0.5 as the detection standard. Moreover, even the accuracy standard of rumor events detection is over 99%, it is nearly 30 hours faster than the official time.

5. Conclusions and future works

Most existing works introduce neural networks for rumor events detection. However, these methods do not consider the aggregation degree of events in the time dimension when constructing the inputs of deep neural networks. And they also ignore the human emotional reactions to events, especially the fine-grained event sentiments, such as happiness, anger, sadness, etc. In this research, we construct a Sentiment Dictionary (SD) including a Sentiment Word Dictionary (SWD) and a Sentiment Emoticon Dictionary (SED) to capture the fine-grained human emotional reactions to different events. Besides, a Two-steps Dynamic Time Series (TsDTS) method based on a fuzzy clustering algorithm and information granule is adopted to retain the distribution information of social events over time. At last, a two-layer cascaded GRU model based on the SD and TsDTS, shorted as SD-TsDTS-CGRU, is proposed in this paper to detect rumor events from Online Social Networks. The experimental results on the real Sina Weibo and Twitter dataset show that our method outperforms six baseline rumor events detection algorithms.

Although the method proposed in this paper outperforms the latest rumor events detection algorithms, there are several limitations or space for improvement. In the first, it is observed that there are certain differences in types of sentiments contained in rumor and non-rumor events from our experiments and the literature [15]. Thus, we will further study the effects of combinations or associations between different sentiments for rumor events detection. Secondly, to reduce the negative impacts of event-independent sentimental information on rumor events detection mentioned in Case-2, an irrelevant information filtering algorithm will be designed in the subsequent research. At last, only the social texts are taken into consideration in this paper at present. In future work, we will consider the more rich text, such as videos, images, hyperlinks, etc.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Zhihong Wang: Investigation, Supervision, Formal analysis, Writing - original draft. **Yi Guo:** Investigation, Supervision, Writing - review & editing.

Acknowledgments

This research is financially supported by The National Key Research and Development Program of China (grant number 2018YFC0807105), National Natural Science Foundation of China (grant number 61462073) and Science and Technology Committee of Shanghai Municipality (STCSM) (under grant numbers 17DZ1101003, 18511106602 and 18DZ2252300).

References

[1] Y. Li, Y.H. Chen, T. Liu, Survey on predicting information propagation in microblogs, J. Softw. 27 (2) (2016) 247–263.
[2] L. Zhang, X.T. Guo, Rumor, credibility and mass incident: based on the research about salt rumors and salt panic, Chin. J. Journal. Commun. (7) (2012) 12–18.
[3] F. Yang, Y. Liu, X. Yu, M. Yang, Automatic detection of rumor on sina weibo, in: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, 2012, p. 13.
[4] X. Liu, A. Nourbakhsh, Q.Z. Li, R. Fang, S. Shah, Real-time rumor debunking on twitter, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM), 2015, pp. 1867–1870.
[5] J. Ma, W. Gao, Z.Y. Wei, Y.M. Lu, K.F. Wong, Detect rumors using time series of social context information on microblogging websites, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM), 2015, pp. 1751–1754.
[6] S. Kwon, M. Cha, K. Jung, Rumor detection over varying time windows, PLoS One 12 (1) (2017) e0168344.
[7] G. Liang, W.B. He, C. Xu, L.Y. Chen, J.Q. Zeng, Rumor identification in microblogging systems based on users' behavior, IEEE Trans. Comput. Soc. Syst. 2 (3) (2015) 99–108.
[8] S. Kwon, M. Cha, K. Jung, W. Chen, Y.J. Wang, Prominent features of rumor propagation in online social media, in: IEEE 13th International Conference on Data Mining (ICDM), 2014, pp. 1103–1108.
[9] K. Wu, S. Yang, K.Q. Zhu, False rumors detection on sina weibo by propagation structures, in: IEEE 31st International Conference on Data Engineering (ICDE), 2015, pp. 651–662.
[10] J. Ma, W. Gao, K.F. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 1, 2017, pp. 708–717.
[11] Z.H. Wang, Y. Guo, Automatic rumor events detection on chinese microblogs, J. Chin. Inf. Process. 33 (6) (2019) 132–140.
[12] E.S. Mao, G. Chen, X. Liu, B. Wang, Research on detecting micro-blog rumors based on deep features and ensemble classifier, Appl. Res. Comput. 33 (11) (2016) 3369–3373.
[13] W.T. Yang, G. Liang, K. Xie, J. Yang, C. Xu, Rumor detection method based on burst topic detection and domain expert discovery, J. Comput. Appl. 37 (10) (2017) 2799–2805.
[14] Q.Y. Wu, L.N. He, Analysis of short text rumors via multi-label bi-term topic model, J. Intell. 36 (3) (2017) 92–97.
[15] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.
[16] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K.F. Wong, M.Y. Cha, Detecting rumors from microblogs with recurrent neural networks, in: The International Joint Conference on Artificial Intelligence (IJCAI), 2016, pp. 3818–3824.

[17] W. Chen, Z. Yan, K.Y. Chai, T.L. Chiew, S.L. Bu, Unsupervised rumor detection based on users' behaviors using neural networks, Pattern Recognit. Lett. 105 (2018) 226–233.
[18] T. Chen, X. Li, H. Yin, Y. Wang, Call attention to rumors: deep attention based recurrent neural networks for early rumor detection, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2018, pp. 40–52.
[19] D.D.B. Vaghela, P. Divya M, Rumor detection with twitter and news channel data using sentiment analysis and classification, Int. J. Adv. Eng. Res. Dev. 5 (02) (2018) 924–930.
[20] H.R. Shou, S.Q. Deng, J. Xu, Detecting online rumors with sentiment analysis, Data Anal. Knowl. Discov. 1 (7) (2017) 44–51.
[21] Z. Yan, W. Chen, K.Y. Chai, Detecting rumors on online social networks using multi-layer autoencoder, in: IEEE Technology & Engineering Management Conference, 2017, 2017, pp. 437–441.
[22] Z. Liu, Z.H. Wei, R.X. Zhang, Rumor detection based on convolutional neural network, J. Comput. Appl. 37 (11) (2017) 3053–3056.
[23] P.T. Jia, H.C. He, L. Liu, T. Sun, Overview of time series data mining, Appl. Res. Comput. 24 (11) (2007) 15–18. 19
[24] H.L. Li, C.H. Guo, Piecewise aggregate approximation method based on cloud model for time series, Control Decis. 26 (10) (2011) 1525–1529.
[25] A.M. Durán-Rosal, P.A. Gutiérrez, A. Carmona-Poyato, C. Hervás-Martínez, A hybrid dynamic exploitation barebones particle swarm optimisation algorithm for time series segmentation, Neurocomputing 353 (2019) 45–55.
[26] R. Palivonaite, K. Lukoseviciute, M. Ragulskis, Algebraic segmentation of short nonstationary time series based on evolutionary prediction algorithms, Neurocomputing 121 (2013) 354–364.
[27] M. Pérez-Ortiz, A.M. Durán-Rosal, P.A. Gutiérrez, J. Sánchez-Monedero, A. Nikolaou, F. Fernández-Navarro, C. Hervás-Martánez, On the use of evolutionary time series analysis for segmenting paleoclimate data, Neurocomputing 326–327 (2017) 3–14.
[28] W.Y. Sun, X.L. Su, Summary of time series segmentation methods, 2012, http://www.paper.edu.cn/releasepaper/content/201202-913.
[29] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, A Practical Guide to Sentiment Analysis, Springer, Cham, Switzerland, 2017.
[30] H. Peng, E. Cambria, A. Hussain, A review of sentiment analysis research in chinese language, Cognit. Comput. 9 (4) (2017) 423–435.
[31] S. Poria, E. Cambria, L.W. Ku, C. Gui, A. Gelbukh, A rule-based approach to aspect extraction from product reviews, in: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), 2014, pp. 28–37.
[32] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl. Based Syst. 108 (2016) (2016) 42–49.
[33] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Frankl. Inst. 355 (4) (2018) 1780–1797.
[34] D. Rajagopal, E. Cambria, D. Olsher, K. Kwok, A graph-based approach to commonsense concept extraction and semantic similarity detection, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 565–570.
[35] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1601–1612.
[36] Y. Ren, D. Ji, H. Ren, Context-augmented convolutional neural networks for twitter sarcasm detection, Neurocomputing 308 (2018) 1–7.
[37] C. Cai, L. Li, D. Zeng, New words enlightened sentiment analysis in social media, in: 2016 IEEE Conference on Intelligence and Security Informatics (ISI), 2016, pp. 202–204.
[38] J.W. Ren, L. Yang, X.F. Wu, Y. Lin, H.F. Lin, Public sentiment trend prediction of microblog events based on affective commonsense knowledge, J. Chin. Inf. Process. 31 (2) (2017) 169–178.
[39] S. Xiong, K. Wang, D. Ji, B. Wang, A short text sentiment-topic model for product reviews, Neurocomputing 297 (2018) 94–102.
[40] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, B. Gupta, Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of arabic hotels' reviews, J. Comput. Sci. 27 (2018) 386–393.
[41] H. Jo, S.M. Kim, J. Ryu, What we really want to find by sentiment analysis: the relationship between computational models and psychological state, 2017, ArXiv preprint arXiv:1704.03407.
[42] R. Ren, D.D. Wu, T. Liu, Forecasting stock market movement direction using sentiment analysis and support vector machine, IEEE Syst. J. 13 (1) (2018) 760–770.
[43] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.
[44] S. Zhang, Z. Wei, Y. Wang, T. Liao, Sentiment analysis of chinese micro-blog text based on extended sentiment dictionary, Future Gener. Comput. Syst. 81 (2018) 395–403.
[45] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, E. Cambria, Learning word representations for sentiment analysis, Cognit. Comput. 9 (6) (2017) 843–851.
[46] H. Peng, Y. Ma, S. Poria, Y. Li, E. Cambria, Phonetic-enriched text representation for chinese sentiment analysis with reinforcement learning, 2019, ArXiv preprint arXiv:1901.07880.
[47] F. Liu, J. Zheng, L. Zheng, C. Chen, Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification, Neurocomputing 371 (2020) 39–50.
[48] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, Y. Jararweh, Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews, Int. J. Mach. Learn. Cybern. 10 (8) (2019) 2163–2175.
[49] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM, in: In Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 5876–5883.
[50] M. Yang, Q. Qu, X. Chen, C. Guo, Y. Shen, K. Lei, Feature-enhanced attention network for target-dependent sentiment classification, Neurocomputing 307 (2018) 91–97.
[51] B. Ghanem, P. Rosso, F. Rangel, An emotional analysis of false information in social media and news articles, 2019, ArXiv preprint arXiv:1908.09951.
[52] W.Y. Li, Research on microblog rumors detection pattern based on sentiment analysis, 2016, Chongqing University.
[53] K.L. Zu, M.W. Zhao, K. Guo, H.F. Lin, Research on the detection of rumor on sina weibo, J. Chin. Inf. Process. 31 (3) (2017) 198–204.
[54] V. Sivasangari, A.K. Mohan, K. Suthendran, M. Sethumadhavan, Isolating rumors using sentiment analysis, J. Cyber Secur. Mobil. 7 (1) (2018) 181–200.
[55] L.H. Xu, H.F. Lin, Y. Pan, H. Ren, J.M. Chen, Constructing the affective lexicon ontology, J. China Soc. Sci. Tech. Inf. 27 (2) (2008) 180–185.
[56] S.M. Mohammad, P.D. Turney, Crowdsourcing a word-emotion association lexicon, Comput. Intell. 29 (3) (2013) 436–465.
[57] S. Fahlman, Wikipedia, https://en.wikipedia.org/wiki/Scott_Fahlman, 2018-9-19.
[58] J.C. Bezdek, Objective function clustering, in: Pattern Recognition with Fuzzy Objective Function Algorithms, 1981, pp. 43–93.
[59] L.A. Zadeh, Advances in Fuzzy Set Theory and Application, North-Holland Publishing Company, Amsterdam, 1979.
[60] L.Z. Wang, X.D. Liu, Time-series segmentation based on fuzzy clustering algorithm and information granules, Fuzzy Syst. Math. 29 (1) (2015) 175–182.
[61] Z. Zhao, P. Resnick, Q. Mei, Enquiring minds: early detection of rumors in social media from enquiry posts, in: International Conference on World Wide Web (WWW), 2015, pp. 1395–1405.
[62] J. Sampson, F. Morstatter, L. Wu, H. Liu, Leveraging the implicit structure within social media for emergent rumor detection, in: ACM International on Conference on Information and Knowledge Management (CIKM), 2016, pp. 2377–2382.

Zhihong Wang was born in Jiangsu, China in 1990. He received the M.S. degrees in Computer Science and Engineering from East China University of Science and Technology, Shanghai, China. He is currently pursuing a Ph.D. degree in Computer Science and Technology from East China University of Science and Technology. His research interests include text mining, machine learning and big data analysis.

Yi Guo received his M.S. degree in Computer Science from Xidian University, Xi'an, China and Ph.D. degree in Computer Science from Heriot-Watt University, Edinburgh, Scotland in 2005. He is currently a Professor at East China University of Science and Technology. His research concentrates on text mining, information extraction, knowledge discovery and business intelligence analysis. He is the member of Committee Board of National Engineering Laboratory for Big Data Distribution and Exchange Technologies and acts as MIEEE MCMI MIET MBCS and APMG-MSP/PRINCE2-Practitioner.