IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 18, 2019, accepted August 2, 2019, date of publication August 12, 2019, date of current version August 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934570

An Automatic Event Detection Method for Massive Wireless Access Prediction

MINGYONG YIN[1,3,4], XINGSHU CHEN[2,3], HAIZHOU WANG[2,3], QIXU WANG[2,3], CHENXI MA[1], AND XUE QIN[5]

[1]College of Computer Science, Sichuan University, Chengdu 610065, China
[2]College of Cybersecurity, Sichuan University, Chengdu 610065, China
[3]Cybersecurity Research Institute, Sichuan University, Chengdu 610065, China
[4]Institute of Computer Application, Mianyang 621900, China
[5]Department of Computing Sciences, Texas A&M University at Corpus Christi, Corpus Christi, TX 78412, USA

Corresponding author: Xingshu Chen (chenxsh@scu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61272447 and Grant 61802271, and in part by the Fundamental Research Funds for the Central Universities under Grant SCU2018D018 and Grant SCU2018D022.

ABSTRACT The scale of mobile users for parallel access is constrained by the capacity of the base stations. When extremely dense terminal access exceeds the capacity of the base stations, access failure and a performance degradation will occur. The early detection and prediction of important events and the timely detection of possible large-scale terminal access are significant aspects in ensuring the quality of the communication achieved. For the automatic detection of events, methods based on a neural network can learn features automatically without feature engineering and have been proven to be efficient for event detection. As is well known, constructing an adequate input vector that can represent sufficient information is a challenge to a neural network-based approach, particularly for problems caused by Chinese word segmentation and too many unknown communication words. To cope with this problem, a novel representation method that combines the different features with word vectors is proposed to deal with the problem of Chinese event trigger identification. We then use a gated recurrent unit network to train and predict the event trigger and carry out comparative experiments on different methods and feature combinations. The experiment results of the proposed model show that the F1 value can reach 84% for the experimental dataset. Furthermore, the combination of lexical and syntactic features with a neural network was proven to be helpful for this task, although the contributions vary in magnitude for different features. Our study provides directions for further research on the use of lexical and syntactic features with a neural network for an event detection task.

INDEX TERMS Trigger identification, GRU, Chinese event extraction, wireless communication.

I. INTRODUCTION

The number of subscribers that a mobile communication base station can accommodate varies based on the differences in carrier frequencies and call loss rates. With the development of communication technologies from 2G to 5G, the capacity of single base station access has increased. However, to avoid a dynamic crush from a flash crowd, automatically providing the appropriate response time to meet the user expectations and supporting an uninterrupted use of communication services remain challenging problems [1], [2].

It is extremely important to detect emergent events and predict large-scale base station access caused by a possible crowd gathering over time, thereby enabling a dynamical increase in the mobile capacity supply, and ensuring the quality of the communication. We can further divide emergent events into two categories, namely, social events that may incur a gathering of people and a failure or security event occurring in a wireless communication infrastructure, the potential causes of which can be found in [31]–[37]. As depicted in fig.1, as a typical scenario, by detecting events from a social network, a communication engineer can predict potential flash crowd events and increase the base station capacity before they occur.

As is well known, the automatic extraction of event information has attracted increasing attention in the field of natural language processing in recent years. Event detection precedes event extraction, and event triggers are words

The associate editor coordinating the review of this article and approving it for publication was Guan Gui.

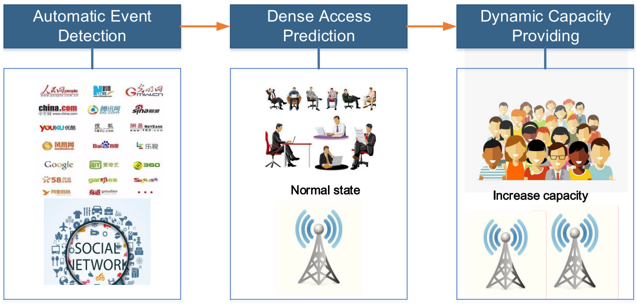

FIGURE 1. **Provisioning of dynamic resources driven by event detection.**

that can clearly indicate the occurrence of an event; thus, the identification process of an event trigger should first be resolved [10], [25]–[27].

An event trigger represents the occurrence of an event, usually a verb or a noun, that can represent an action or change of state. For example, the word "detect" in the candidate event sentence "Unknown wireless signal repeatedly detected in Chengdu Shuangliu International Airport" as the trigger is composed of verbs, and in the candidate event sentence "175 thousand wireless cameras have serious security vulnerabilities," the trigger "vulnerability" is a noun. Chinese trigger identification is due to certain language characteristics expressed in Chinese, as well as the influence of Chinese word segmentation, which leads to a difficulty regarding trigger extraction, such as an inaccurate word segmentation of a trigger, or certain event instances with a lack of relevant information because of too many omissions.

Classic machine learning methods realize Chinese event trigger identification and classification by constructing the features of a trigger, commonly used features including part-of-speech tagging (POS), named entity recognition (NER), and dependency parsing (DP), among others. However, problems such as a loss of semantic information, feature construction, and selection complexity occur. Although different features can be used for a trigger identification task, the way in which different types of feature combinations affect Chinese event trigger identification needs to be further studied.

Along with research advancements in neural networking, a recurrent neural network (RNN) is used to capture a greater semantic meaning of a sentence and has achieved good results in trigger identification without a feature construction; however, this does not mean the feature is not in favor of this task, and thus this paper proposes a new trigger identification model for combining the lexical and syntactic features with a neural network; in addition, a classic feature engineering based machine method is compared with a neural network based method.

In this study, an event trigger identification model was constructed using a recursive neural network combined with multiple feature information for knowledge representation, and a comparative analysis was conducted to show the effectiveness of the proposed model as compared to classic machine learning methods. The contributions of this paper are summarized as follows:

- A novel semantic representation method that combines different features with word vectors is proposed to deal with the problem of Chinese event trigger identification. According to the literature, this is the first study to focus on automatically identifying Chinese events in wireless communication.
- Comparative experiments were carried out using other typical event trigger identification methods and different feature combinations using an open corpus, which can provide a comprehensive reference for other types of Chinese event detection.
- An impact analysis of different feature combinations is provided both qualitatively and quantitatively, which indicates areas for further research on how to use lexical and syntactic features with a neural network in an event detection task.

The remainder of this paper is composed as follows: In section 2, we describe previous studies related to trigger identification. In section 3, we provide a brief introduction regarding semantic representation and a gated recurrent unit (GRU) neural network model. In section 4, we describe our trigger identification framework in detail. In section 5, we describe the applied datasets and experiments conducted, and present the experiment results of the proposed method based on our model and traditional machine learning techniques with different feature combinations. Finally, in section 6, we present some concluding remarks regarding the application of the trigger identification model.

II. RELATED WORK

A. STATISTICS AND RULES-BASED METHOD

Regarding the automatic detection of events for the communication domain, to the best of our knowledge, little specialized research has been conducted. Traditionally, in the area of event detection, two methods are usually adopted to extract an event trigger: methods based on statistics and methods based on rules.

A method based on statistics operates on the statistical processing of a large-scale corpus, such as in [3], [28]. Using statistical tools, Fu *et al.* found that event triggers mainly include nouns, verbs, and gerunds. Such a method based on statistics relies heavily on experimentation, and it is generally believed that, once the corpus is sufficient and representative, it can obtain reliable statistical results. However, owing to the nonergodicity of statistics led by a limited volume of the corpus, the performance is in fact restricted. A rule-based method is a theoretical analysis approach, and deliberately constructed rules can represent linguistic phenomena under specific conditions; therefore, this method can be extremely straightforward. For a rule generation, however, expert involvement is required, and when considering the diversity and openness of a linguistic phenomenon, this method can only work in a specific language environment.

B. CLASSIC MACHINE LEARNING METHOD

For classic machine learning methods, the classifier needs to be trained by selecting the effective features first, and then converting the trigger identification into a classification problem. Ahn [4] combined a method based on the k-nearest neighbor algorithm (MegaM) and a method based on the maximum entropy algorithm (TiMBL) into two machine learning algorithms, which respectively conduct event-type recognition and event element recognition. The recognition features used include lexical features and dictionary features, and the experiment results show that the performance of the proposed method outperforms a method using a single algorithm, although an imbalance between the positive and negative examples and a data sparseness may occur. Llorens *et al.* [5] use a conditional random field to learn the language features, event recognition, and event role labeling, thereby improving the system performance. Zhao *et al.* [6] proposed a method based on a combination of trigger expansion and binary classification to identify the event categories. In addition, multi-class classification models are used to identify the event elements, avoiding an imbalance between positive and negative examples during the process of trigger identification. Hao *et al.* [20] use a dependency analysis method to mine the syntactic relationship between a trigger and other words, combining the dependency characteristics and part-of-speech features for trigger identification. Fu *et al.* [19] combine a trigger vocabulary and word features to construct a support vector machine (SVM) classifier for identifying anomalous events, which is highly accurate regarding the identification of attacks, injuries, deaths, and criminal arrests. Chen and Ji [21] pointed out that the problem of Chinese word segmentation may cause a trigger to consist of multiple words, or a trigger may be a single word. The contribution of lexical, syntactic, and semantic features in event category recognition and event element recognition was also discussed.

Compared with template-based matching methods, machine learning-based methods are simple and efficient, and require less domain knowledge; however, they still have deficiencies that need to be solved. The disadvantages of a machine learning feature engineering method can be summarized as follows:

- The selection of features depends on the specific domain and language.
- The features are usually generated using natural language processing (NLP) tools, and once the extracted features are insufficiently accurate, an error propagation will occur, thereby affecting the identification performance.
- The generation of certain complex features often requires the development of experts with extensive linguistic knowledge.

C. DEEP LEARNING METHOD

In recent years, deep learning methods have achieved good results in the field of NLP, including text classification and sentiment analysis. In the field of event extraction, many researchers have used deep neural network models for event extraction, and convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are often adopted. Neural network-based methods generally take an event extraction task as a sequence annotation. Zhong *et al.* [18] proposed an event recognition method based on a CNN with domain self-adaptation, which can automatically learn features from sentences, avoiding the dependence on NLP tools in feature engineering and reducing the error propagation. Chen *et al.* [7] introduced word vectors to calculate rich semantic information in sentences. Considering that, although a CNN can extract the most important information from a sentence, it may miss other important information in sentences containing multiple events, they proposed a DMCNN method to tackle this problem. Nguyen *et al.* [8] combined the advantages of a method proposed in [7], [9], presenting a joint event extraction model based on an RNN and obtaining good experimental results that support multiple event detection concurrently. Feng *et al.* [11] proposed a language-independent neural network model that can identify and classify event triggers simultaneously, and conducted experiments on English, Chinese, and Spanish corpora proving the efficiency of the model.

In terms of Chinese event extraction, Zeng *et al.* [12] proposed a convolution-bidirectional long-short term memory (BiLSTM) neural network model for detecting Chinese events, and discussed the language characteristics caused by word segmentation in Chinese event extraction, focusing on word-based vector information representation and comparisons. Zhang [13] proposed a joint event extraction framework that combines a window-based convolutional neural network and a recurrent neural network. It realizes the identification of an event trigger and the extraction of event elements at the same time, avoiding the problem of error propagation.

To summarize, neural-network based methods are being increasingly adopted owing to their innovative semantic representation capability, and because they make full use of computing resources for optimization through training and testing. However, promoting the representation capability in a specific domain and achieving good results in practical applications remain challenging tasks, particularly for automatic Chinese event detection. For instance, how to deal with mistakes incurred by a word segment error, how to improve the semantic representation capability, and whether language features can be used to promote the trigger identification accuracy still need to be further researched.

III. SYSTEM MODEL AND DESIGN GOALS

A. MODEL DESCRIPTION

For an automatic resource supply driven through an event detection, the critical part is an accurate event trigger identification. We designed the model to include two aspects: 1) a distribution representation of Chinese events used to

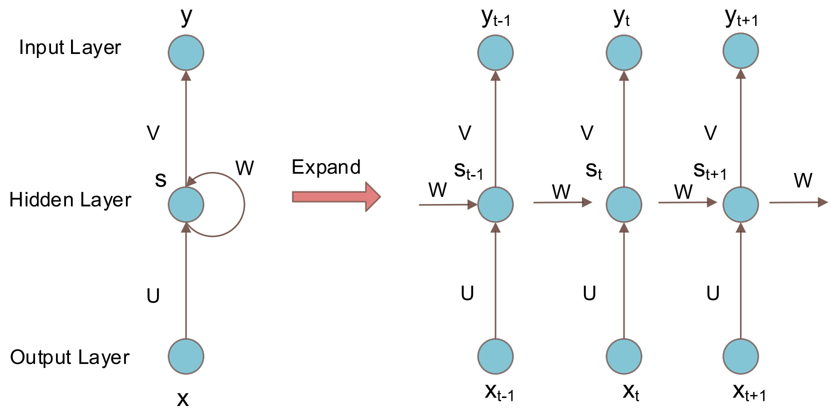

FIGURE 2. **Structure diagram of the recurrent neural network.**

construct semantical representation vectors through word embedding, combined with multiple morphological and syntactic features, and 2) an event trigger detection model, which is a classical sequential annotation problem through which we need to use the context information before and after the target word. Therefore, by constructing a two-way information input, using the context memory capability and fast learning capability of a GRU neural network, we can learn the optimal trigger word detection model based on the training datasets. In the latter part of this section, we provide a brief introduction regarding the two basic models that are critical for such a task, namely, the distributed representation model and the GRU model. The distributed representation model mainly describes the method for converting original information into a format that is feasible for computer processing. The GRU model mainly introduces the learning and decision-making processes of a task. Finally, we end this section with the details of the design goals.

B. DISTRIBUTED REPRESENTATION

To facilitate the computer processing of wireless communication related events, it is necessary to convert the information into a numerical symbol that the computer can recognize and calculate, and a word vector is usually used as a method for converting words into real numbers.

In traditional statistical-based natural semantics processing methods, words are usually regarded as an atomic symbol. Each word is represented by a long vector, the length of which is the size of the dictionary, where only one dimension has a value of 1, and the remaining values are all zeros. This representation method is called one-hot representation.

There are two problems with this representation method: The first is a "lexical gap" in which any two words are isolated and the similarity of the two words cannot be expressed. The other is the fact that the vector dimension of this representation method is usually extremely large, which may cause a dimensional problem when solving section tasks, such as the building of a language model [23].

A word vector is a low-dimensional real vector with a distributed representation. Through training, words are mapped into a fixed length vector, and all word vectors are put together to form a word vector space, with each word being a point in the space. Compared with a one-hot representation, the word vectors generated through a distributed representation are smaller in dimension and contain rich semantic information, with the distance between word vectors representing the similarity between them. The Word2vec model proposed by Mikolov *et al.* [15] is a common training method used for word vectors, which includes continuous bag-of-words (CBOW) and skip-gram models. A skip-gram model predicts the context words given the target word, whereas the CBOW model predicts the target word given the context words. Compared with other neural network models, the CBOW and skip-gram models remove the hidden layer. This simplified strategy makes it more efficient, and allows word vectors to be trained on a larger corpus scale.

C. GATE RECURRENT UNITS NEURAL NETWORK

In general, we can treat a trigger identification as a sequence annotation task, in which the input is the encoded sentences, and the output is the annotation result, which contains information regarding whether the word is an event trigger. For a sequence annotation, a recurrent neural network model is always considered a candidate method. RNNs are a type of neural network that are generally used to process sequence data.

Unlike traditional feed forward neural networks, an RNN introduces loops that can store information while processing new inputs, making them ideal for processing sequence data. The output of the current time in the sequence is related to the information of the input layer and the output of the previous time in the sequence. That is, the RNN can memorize and apply the information of the previous moment. The states of the previous hidden nodes can be passed, and the input of the hidden layer includes the output of the input layer and the output at the previous moment, as shown in fig.2.

Unfortunately, when the distance from the target word widens, the learning dependencies become more difficult because of the gradient vanishing or exploding problem. During the training of a deep network, when the gradients are being propagated back to the initial layer following the chain rule, the gradients coming from the deeper layers have to go through continuous matrix multiplications, and as they approach the earlier layers, their value will exponentially decrease until they become too small for the model to learn, which is considered a vanishing gradient problem. In the same way, if the initial value is large, they become larger and eventually blow up, making them impossible to calculate, which is the exploding gradient problem.

When gradients explode, gradient clipping is a solution used to fix this problem by using a predefined threshold on the gradients to prevent it from becoming too large; in this way, the direction of the gradients is not changed.

A popular and widely used solution is the gated recurrent unit architecture, which is designed using an exquisite network structure to alleviate the problem of gradient vanishing [5], [9], [14]. Gated recurrent units are a gating mechanism introduced in reference [24]. They can learn long-term dependencies with fewer parameters and have a higher computational efficiency, thereby achieving success in many natural language processing tasks. We denote the following: input vector X_t, output vector Y_t, update gate vector z_t, and reset gate vector r_t. In addition, W, U, and b are parameter matrices and a bias vector, and the activation function is denoted as σ_g.

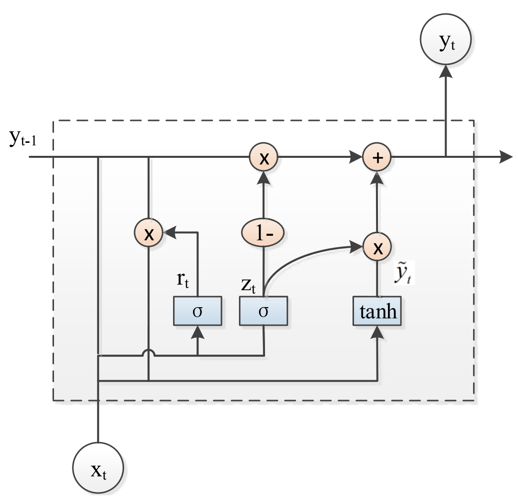

FIGURE 3. Process diagram of the gate recurrent network.

As shown in fig.3, the first step in the GRU loop module is applying an update gate to observe and decide how much information should be passed to the next state.

$$z_t = \sigma_g(W_z x_t + U_z y_{t-1} + b_z) \tag{1}$$

The second step is a reset gate to observe and judge the importance of new memory units.

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

The third step is to generate a new memory unit, which is generated by the hidden state of the previous moment and the input of the current moment. That is, this stage can reasonably merge the newly observed information with the history of hidden states.

$$y_t = \tan_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{3}$$

The last step is to generate a new hidden state.

$$y_t = (1 - z_t) \circ y_{t-1} + z_t \circ y_t \tag{4}$$

D. DESIGN GOALS

The design goal is to develop an event driven automatic resource scheduling framework based on the model introduced above. Specifically, we aim to achieve the following objectives.

1) Rich semantic representation: This representation method should be universal for different types of information, and able to capture the meaning of words, maintaining as much information of the sentences as possible, such as the sequence order of words.
2) Accurate event detection: This method should be able to learn fast, and the precision and recall rate of the annotation result should be sufficient for the automatic detection of wireless communication related events.
3) Predictable event classification: This model needs to support efficient event classification to provide information for further analyzing the possibility of crowds gathering.

IV. EVENT TRIGGER IDENTIFICATION FRAMEWORK

A trigger identification task is usually divided into two parts, the detection of the trigger and its classification. If a trigger is incorrect during the detection process, its detection will directly affect the classification process of the actual trigger, resulting in an error propagation of the subtasks.

In this study, a trigger identification framework based on a GRU neural network is proposed. The input is a candidate event sentence, and the output is the trigger word annotation and trigger classification results.

The processing flow for the event trigger identification task in this paper is shown in Figure 3, which is described as follows:

- First, the word embedding is trained using the CEC corpus, and sentence and word segmentation are conducted on the text in the dataset. Sentence segmentation is applied to divide the text according to punctuation marks such as ".","?," and "!," and the word segmentation divides each sentence into a sequence of words. The word embedding is then trained using text segmented for a distributed representation.
- Second, an input vector is constructed by applying part-of-speech tagging, named entity recognition, dependency parsing, and the concatenation of the word and feature vectors of each word sequence as events triggering an identification of the model input.
- Third, based on the target vector acquired in the training set, the event trigger identification model is trained, and the trigger of the test data is predicted.

TABLE 1. Part-of-speech tags and meanings.

ID	Label	Dscreption	ID	Label	Dscreption
0	nt	time noun	13	c	conjunction
1	m	number	14	nl	place noun
2	wp	punctuation	15	nh	person noun
3	n	common nouns	16	j	abbreviations
4	ns	geographic nouns	17	ws	foreign word
5	nd	direction noun	18	b	noun modifier
6	q	quantifier	19	r	pronoun
7	v	verb	20	i	idiom
8	p	preposition	21	z	descriptive words
9	u	auxiliary verb	22	k	suffix
10	d	adverb	23	ni	organization name
11	nz	proper nouns	24	o	onomatopoeia
12	a	adjective			

A. FEATURE VECTOR CONSTRUCTION

This study regards event trigger identification as a sequence labeling task and uses a GRU model for trigger detection. In addition to the use of word vectors, the proposed model combines some syntactic features and lexical features that are helpful for a trigger extraction, including the part-of-speech, named entity, and dependency parser features.

After segmentation, the sentences are represented by a sequence of words, $W = w_0, w_1, w_2, \ldots, w_{n-1}$, where n is the length of the sentence and w_i represents the *ith* word in the sentence. In an event trigger detection task, it is necessary to judge whether each word is a trigger, and if so, it is necessary to predict the type of the event represented by the trigger. The process of constructing the eigenvectors is a process of encoding sentences, and transforming the semantic information and grammatical features contained therein into a real vector. A word vector is a way to convert words in a sentence into a real number vector that can be calculable, which contains a wealth of semantic information. In this study, the skip-gram model in word2vec and the improved method of hierarchical Softmax proposed by Mikolov *et al.* [15] are used to train all words in the CEC corpus, and a word vector representation of words in the corpus is obtained. Each word w_i in the word sequence is represented by a word vector as $[v_0, v_1, v_2, \ldots, v_{m-1}]$, where m is the dimensional size of the word vector.

In addition to a basic word vector, the input vector of the model in this study also adds the part-of-speech vector, named entity vector, and syntax vector features.

1) PART-OF-SPEECH FEATURE VECTOR

The part-of-speech feature is derived from the part-of-speech tagging, and the target is to annotate the words in the given sentences with a verb, a noun, an adjective, or other word type. Part-of-speech tagging is a typical sequence annotation task, in which words are tagged based on the contextual information of the word, including the previous and subsequent words and their past-of-speech tags. A trigger is a word that indicates the occurrence of an event, usually a verb or a noun, and the probability that a word labeled as a verb or a noun will be recognized as a trigger will increase.

In this paper, the LTP tool is used to conduct part-of-speech tagging of word sequences after word segmentation citeb21. The resulting format of the part-of-speech tagging is $[nt, nt, nt, nt, nt, wp, ns, ns, v, m, q, n]$. The part-of-speech tag and its meaning are shown in table 1. Finally, the part-of-speech tag is converted into a part-of-speech vector. The tag set used in this study includes 25 types of speech, and a part-of-speech tag dictionary with a length of 25 can be constructed. Each word corresponds to a vector, and only one bit of the vector value is set to 1, corresponding to the position of the part of the word in the portion of the lexical dictionary, and the remaining position is zero. The part-of-speech feature vector corresponding to the hypothetical word w_i is $[pos_0, pos_1, \ldots pos_j, \ldots pos_{24}]$, and the setting method of pos_j is as shown in equation (5):

$$pos_j = \begin{cases} 1, & \textit{if the POS ID of } w_i \textit{ is } j. \\ 0, & \textit{otherwise}. \end{cases} \tag{5}$$

2) NAMED ENTITY FEATURE VECTOR

The named entity feature is derived using named entity recognition, which refers to words with special meaning in a text, including proper nouns such as people, places, and institutions. The named entity features affect the identification of event triggers. In this study, the LTP tool is used to mark the named entities in the word segmentation process, and the BIEO tagging mode is used to identify the names of people, places, and institutions in the sentence. The named entity type labels and their meanings are shown in table 2. Finally, the named entity features are converted into named entity vectors. From the above table, a named entity tag dictionary with a length of 13 is obtained. Each word corresponds to a vector of length 13, and only one bit is set to 1, and the position of the 1 corresponds to the tag in the named entity tag dictionary. The hypothetical word corresponding to the named entity feature vector $[ner_0, ner_1, \ldots, ner_j, \ldots, ner_{12}]$, ner_j is set as shown in equation (6):

$$ner_j = \begin{cases} 1, & \textit{if NER ID of } w_i \textit{ is } j. \\ 0, & \textit{otherwise}. \end{cases} \tag{6}$$

TABLE 2. Named entity tags and meanings.

ID	Tag	Meaning
0	O	The word is not a named entity
1	B-Ns	Begin of place name
2	E-Ns	End of place name
3	S-Ns	Place name
4	I-Ns	Middle of place name
5	B-Ni	Begin of institution name
6	I-Ni	Middle of institution name
7	E-Ni	End of institution name
8	S-Ni	Institution name
9	B-Nh	Begin of human name
10	E-Nh	Middle of human name
11	I-Nh	End of human name
12	S-Nh	Human name

3) DEPENDENCY FEATURE VECTOR

The concept of dependency parsing describes the asymmetrical binary lexical relationship in the syntactic structure. The dependency relationship consists of two parts: the center word and the dependent word. The dependency relationship represents the semantic dependence between the core word and the dependent word. In simple terms, the dependency analysis identifies the grammatical components such as "subjective" and "fixed complement" in the sentence and obtains the relationship between the components.

TABLE 3. Dependency parsing labels and meanings.

ID	Tag	Meaning
0	ATT	attribute
1	ADV	adverbial
2	RAD	right adjunct
3	IS	independent structure
4	CMP	complement
5	POB	preposition-object
6	COO	coordinate
7	VOB	verb-object
8	SBV	subject-verb
9	HED	head
10	FOB	fronting-object
11	LAD	left adjunct
12	DBL	double
13	IOB	indirect-object

In this paper, the LTP tool is used to analyze the dependency relationship in the word sequence after word segmentation, and a dependent parsing tree is obtained. The dependency syntax type tag and its meaning are shown in table 3. Finally, the dependency features are converted into two feature vectors. One is the parent node vector that depends on each word. Each word corresponds to a vector whose length is 1 plus the length of the sentence. If the current word w_i depends on the word w_j, then the position of word w_j in the vector is set to 1, and the remaining position is set to 0. Assuming that the parent node vector upon which the word depends is $[f_0, f_1, \ldots, f_j, \ldots, f_n]$, f_j is set as shown in equation (7):

$$ner_j = \begin{cases} \textit{if the dependent parent node of } w_i \textit{ is } w_j. \\ 0, \textit{otherwise}. \end{cases} \tag{7}$$

TABLE 4. Event type label and meaning.

ID	Tag	Meaning
0	other	other event type
1	action	about action
2	stateChange	about state change
3	emergency	about emergency event
4	movement	about movement
5	operation	about operation
6	statement	about statement
7	perception	about perception

The other feature vector is a dependency type vector. There are 14 known dependency types, and a dependency dictionary with a length of 14 can be constructed. Each word corresponds to a vector of length 14, with only one bit set to 1, the position of which is based on the order number of the dependency type of the corresponding word in the dependency dictionary, and the remaining position is set to 0. Assuming that the word w_i corresponds to the dependency feature vector $[dep_0, dep_1, \ldots, dep_j, \ldots, dep_{13}]$, which was set as shown in (8):

$$ner_j = \begin{cases} 1, & \textit{if the dependency type ID of } w_i \textit{ is } j. \\ 0, & \textit{otherwise}. \end{cases} \tag{8}$$

B. CONSTRUCTION OF TARGET VECTOR

The target vector is the output target of the framework. In the trigger identification task, the format of the target vector should be $[length, 8]$. This should be consistent with the length of the input vector, that is, the length of the word sequence of the sentence; here, 8 is the vector length corresponding to each word, representing seven types of events and one other type, where "other" indicates that the word is not a trigger. The event-type labels and their meanings are shown in table 4. Assuming that the event type target vector of the word w_i is $[tri_0, tri_1, \ldots, tri_j, \ldots, tri_n]$, tri_j is set as shown in (9):

$$ner_j = \begin{cases} 1, & \textit{if the trigger type ID of } w_i \textit{ is } j. \\ 0, & \textit{otherwise}. \end{cases} \tag{9}$$

C. CHINESE EVENT TRIGGER IDENTIFICATION

The Chinese event trigger identification framework is divided into three layers, namely, an input layer, a hidden layer, and an output layer. Here, the word vector and the above three feature vectors are concatenated as a vector representation of each word in the word sequence, which is the code conversion of the word sequence $W = w_0, w_1, w_2, \ldots, w_{n-1}$ into the real vector $X = x_0, x_1, x_2, \ldots, x_{n-1}$, and the real vector $X = x_0, x_1, x_2, \ldots, x_{n-1}$ was used as the input of the recurrent neural network. We choose GRU as the hidden layer of the framework.

In an RNN, the sequence information is propagated in one direction, and the hidden layer state S_t contains the 0 t information in the sequence. However, this one-way mechanism does not involve the information t $(n-1)$ in the following text.

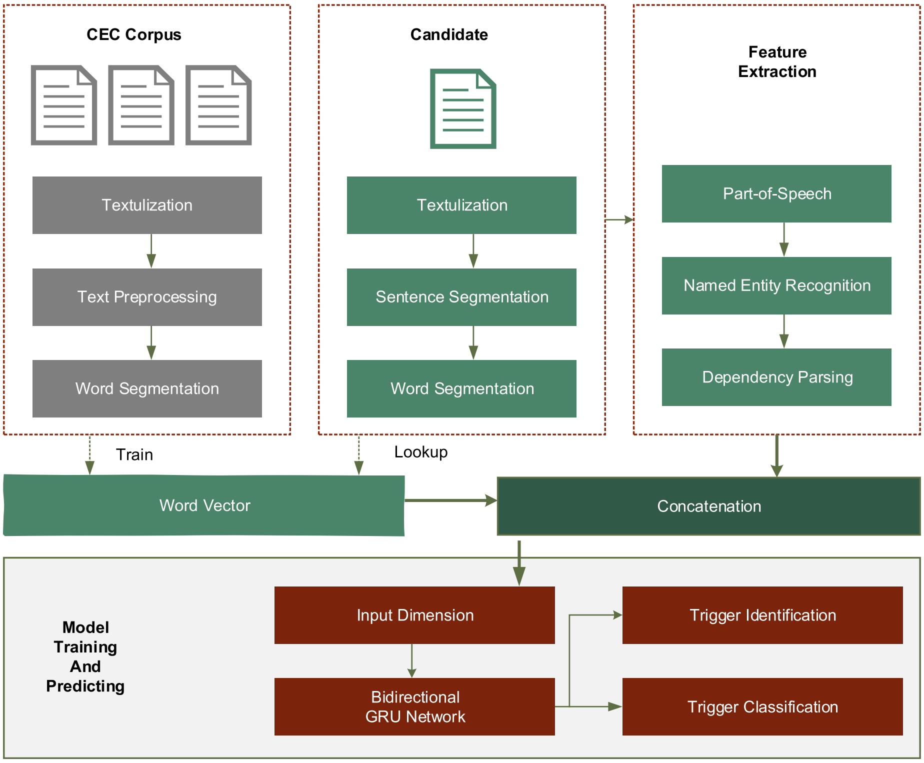

FIGURE 4. Trigger identification flowchart.

The bidirectional recurrent neural network proposed on this basis can solve this problem well. The hidden layer contains both a forward GRU and a reverse GRU. The forward GRU captures the above feature information, and the reverse GRU captures the following information. After the merging of the feature information and the captured context feature information, we finally obtain the global context information, as shown in (10),(11):

$$GRU(x_0, x_1, x_2, \ldots, x_{n-1}) = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}) \quad (10)$$

$$GRU(x_{n-1}, \ldots, x_2, x_1, x_0) = (\alpha'_{n-1}, \ldots, \alpha'_2, \alpha'_1, \alpha'_0) \quad (11)$$

The output layer is a feed-forward neural network. Using Softmax as the activation function, the class probability distribution of each word can be obtained as $P^t_t ri = F^t ri(y_t)$, and the category of event triggers is predicted according to the probability distribution, as shown in (12).

$$y_t = [\alpha_t, \alpha'_t] \quad (12)$$

The event trigger identification framework is shown in fig.4. The event description sentence shown in the figure is "Chengdu and other places in Sichuan also have similar attack." We used the Keras deep learning framework to implement the neural network model, as shown in fig.5. The loss function used in the compilation process is categorical_crossentropy, and the optimizer applied is Adam. The hyperparameter settings for each layer are shown in table5.

V. EXPERIMENTS

A. WORD SEGMENTATION ERRATA

We use the corpus CEC for our experiments, which contains 5,954 triggers. After using the Jieba [16] and LTP [17] word segmentation tools, the event sentences in the CEC corpus are processed separately, and the word segmentation results are compared with the triggers provided in the CEC corpus. The statistics on the specific quantity are shown in table 6.

To handle inconsistencies between the trigger and word segmentation results, we use the global errata method to record all inconsistencies in the training corpus. This scheme can alleviate the inconsistencies between the trigger and word segmentation results in the trigger identification. In this experiment, the LTP tool is used for word segmentation. Typical Chinese word segmentation errors include intersection ambiguity and combinatorial ambiguity, the former of which refers to a string that can be cut in two or more ways, each of which is correct, and the latter refers to combinatorial ambiguity, which may lead a complete trigger word being divided into two independent Chinese characters. The global word segmentation error is corrected through a manual check to avoid a trigger word recognition error caused by a word segmentation.

B. EXPERIMENT ENVIRONMENT, DATASET, AND EVALUATION CRITERIA

We built our implementation on the Keras library of neural networks, and use the Tensorflow as back-end library for

TABLE 5. Hyperparameters used in model training.

Layer	Hyperparameters	Value
Input Layer	Word vector	128
	Part of Speech Vector	25
	Named entity vector	13
	Dependency parse vector	117 , 14
GRU	Output dimension	64
	Merge method	*concat*
	Dropout	0.5
Output Layer	Output dimension	8
	Activation function	*softmax*

The word vector is set to 128 dimensions, the part of speech vector has 25 dimensions, and the named entity vector has 13 dimensions. One of the dependency vectors is the parent node vector that depends on it, where the maximum length of the sentence expected by the CEC after the word segmentation is 116. The second dependent vector is the dependent-type vector, the optimal value of which is obtained after repeated training. The target vector dimension is 8, and the softmax activation function is suitable for multiple classifications.

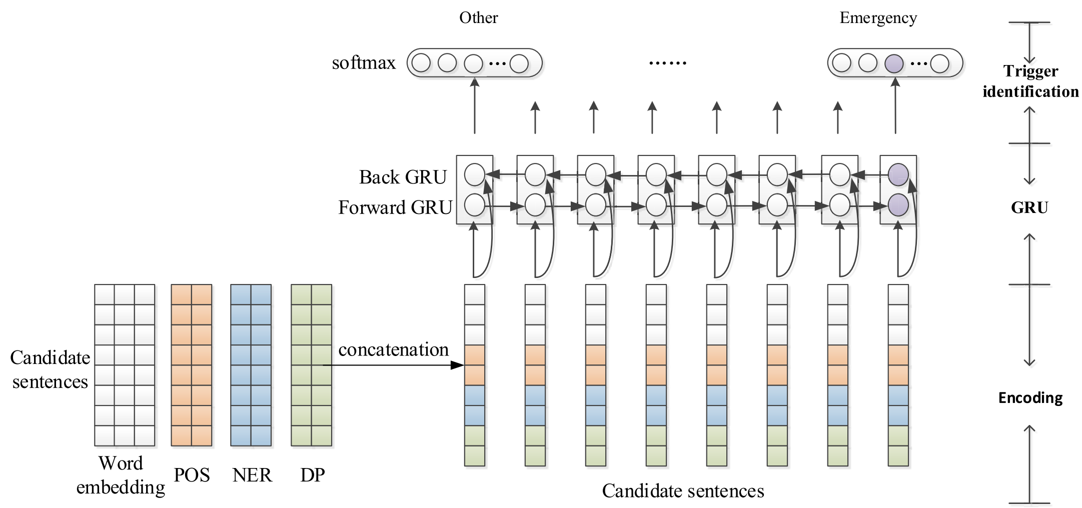

FIGURE 5. Schematic diagram of the event trigger identification model.

TABLE 6. Statistics on the inconsistency of the trigger and word segmentation results in CEC.

Segmentation	Cross-word	Inside-word	Total
Jieba	311	619	930
LTP	691	104	795

Keras [29]. The word embedding tool applied is Gensim [30]. The experiments were conducted on a computer with an Intel Core i7 processor, 16 GB of DDR4 memory, and running macOS 10.13.6.

This study used the CEC corpus as the experiment data, which includes earthquakes, fires, traffic accidents, terrorist attacks, food poisoning, and five other types of emergencies, with a total of 332 events mentioned. Each type of emergency corpus is divided into a training set, verification set, and test set, classified at a ratio of 8:1:1, namely, 267, 33, and 32 events, respectively. The results of the classification of each category are shown in fig.6.

This paper adopts the evaluation criteria widely used in the field of natural language processing to evaluate the

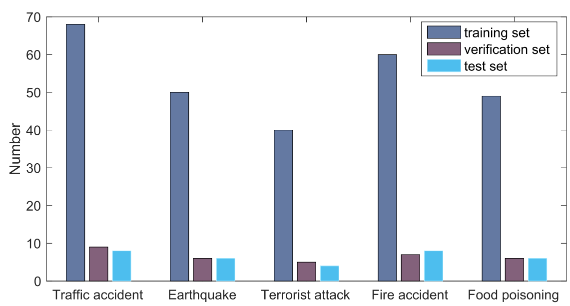

FIGURE 6. CEC dataset partion.

experiment results, namely, the precision, recall, and F-measure, whereas the number of triggers correctly recognized or falsely recognized are abbreviated as TP or FP specifically, and FN indicates the number of missed triggers. Their specific calculation formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{13}$$

TABLE 7. Comparison with different feature combinations.

Method	Trigger detection			Trigger classification		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
GRU+word embedding	0.8102	0.6231	0.7044	0.7054	0.5425	0.6133
GRU+ word embedding + POS	**0.8707**	0.7829	0.8244	**0.7366**	0.6608	0.6866
GRU+ word embedding +NER	0.8597	0.6316	0.7282	0.7284	0.5340	0.6162
GRU+ word embedding + DP	0.8547	0.7560	0.8023	0.7192	0.6348	0.6744
GRU+ word embedding + POS,NER,DP	0.8553	**0.8275**	**0.8413**	0.7111	**0.6893**	**0.7001**

The word vector is set to 128 dimensions, the part-of-speech vector has 25 dimensions, and the named entity vector has 13 dimensions. One of the dependency vectors is the parent node vector that depends on it. where the maximum length of the sentence expected by the CEC after the word segmentation is 116. The second dependent vector is a dependent-type vector, the optimal value of which is obtained after repeated training. The target vector has 8 dimensions, and the softmax activation function is suitable for multiple classifications.

$$R = \frac{TP}{TP + FN} \tag{14}$$

$$F = \frac{2P \times R}{P + R} \tag{15}$$

We set the accepted criteria for a trigger identification as follows: the trigger appeared at the correct position of the word sequence, and the type of the event was correctly identified.

C. COMPARATIVE EXPERIMENTS BY COMBINING MULTIPLE FEATURES

To conduct the event trigger identification experiments, this section combines the GRU recurrent neural network; part-of-speech features, called entity features; and a dependency parser. The experimental results are shown in table 7.

It can be seen from the above experiment results that, when using only the GRU recurrent neural network and the word vector for trigger detection and classification, the obtained result has a better accuracy, although the recall rate is poor, resulting in a low F1 value. After adding the part-of-speech feature, compared with the GRU + word vector, the accuracy of the trigger detection and trigger classification is improved, the recall rate is significantly improved, and an F1 value with a good effect is achieved.

The reason for the improvement can be interpreted as follows: the trigger is a word indicating the occurrence of an event, usually a verb or a noun. When the part-of-speech feature is added, the probability that the verb phrase and the noun phrase are recognized as a trigger is increased, thereby improving the accuracy and recall rate of the trigger identification.

After adding the named entity feature, less improvements in the accuracy, recall rate, and F1 value are achieved than in the GRU + word vector model, which means that the named entity feature contributes little to the trigger identification. After adding the dependency parser feature, compared with the GRU + word vector model, the accuracy of the trigger detection and trigger classification is improved, the recall rate is significantly improved, and the F1 value is achieved. This shows that the dependency characteristics also play an important role, with the dependency characteristics being a syntactic feature of a sentence, reflecting a deeper relationship of the sentence structure than the part-of-speech tagging, and that they have an important influence on improving the accuracy and recall rate of the trigger identification.

TABLE 8. Event trigger recognition results based on event type.

Event type	Detection Result		
	Precision	Recall	F1-Measure
action	0.5035	0.6698	0.5749
stateChange	0.6706	0.7403	0.7037
emergency	0.8302	0.8	0.8148
movement	0.8710	0.6	0.7105
operation	0.8169	0.5743	0.6744
statement	0.9804	0.8621	0.9174
perception	0.7273	0.5333	0.6154

This section describes the part-of-speech feature, the named entity feature, and the dependency feature applied to conduct the comparative experiments, in which the effectiveness of each feature on the event trigger extraction is verified. Finally, in this section, three features are added to the model for training. The obtained event trigger detection and F1 value of the classification results are all optimal.

D. ANALYSIS OF EXPERIMENT RESULTS FOR DIFFERENT EVENT TYPES

Using the model described herein, the accuracy, recall, and F1 value of the trigger identification results of each

TABLE 9. Comparison with other methods using CEC dataset.

Method	Trigger detection			Trigger classification		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
SVM	0.7	0.66	0.68	0.60	0.56	0.57
LSTM	0.8778	0.7719	0.8215	0.7307	0.6411	0.683
GRUA-TRI	0.8553	0.8275	0.8413	0.7111	0.6893	0.7

category are obtained. The experiment results are shown in table 8. It can be seen from the above experiment results that a statement-type event has the best effect, the F1 value of which reaches more than 90%, followed by an emergency-type event, the F1 value of which reaches 80% or more. An action-type event has the worst event recognition effect, the F1 value of which is only 57%.

A statement event is a statement expression event, and the type of event trigger is also relatively simple. In a news text, the common triggers of such events include "report," "propose," "speak," and "declaration," some of which are statements or expressed views. An emergency event is a sudden incident, which indicates the occurrence of certain emergencies. In news reports, the common triggers for such incidents include "earthquake," "seismic," "fire," "tsunami," and "terrorist attack."

The results of the comparative experiment with different event types are consistent with common sense, and the difference can effectively be described as follows: a lack of sufficient training corpus for training each event type, the diversity of trigger words for different event types (typically, there are fewer triggers for a statement-type event than for an action-type event), and, being affected by special Chinese expressions and inaccurate segmentations, a possibly misidentified trigger.

E. COMPARISON EXPERIMENT WITH OTHER MACHINE LEARNING METHODS

This section compares the triggering word recognition method based on a neural network using a traditional machine learning method. With a neural network method, the LSTM neural network and the GRU neural network unit are respectively used, and the same feature vector is constructed. In the machine learning method, the event trigger identification classifier is constructed according to the method by Ameera *et al.* [1], and the part-of-speech feature, called the entity feature, and the dependency relationship feature are used. The contextual characteristics of the word (including three words on the left side of the candidate event trigger and all its features, and two words on the right side of the candidate element and all its features). A study by Zhong [18] shows that the first three words and the last two words below can provide more than 90% of the word information. Next, the SVM classifier is used to detect and classify event triggers. The experiment results are shown in table 9, in which the GRUA-TRI is the trigger identification model based on the GRU recurrent neural network proposed in this paper.

It can be seen from the above experiment results that a trigger detection and trigger classification using the LSTM recurrent neural network and GRU recurrent neural network can outperform the traditional machine learning method, which shows that a neural network method is an optimal choice for an event trigger identification task. Among them, a GRU neural network can achieve a better F1 value, reaching 70%. The trigger detection method based on a neural network not only avoids a lot of manpower, it also does not need particularly complex features to be designed. The algorithm has good portability and does not depend on specific fields. Moreover, the model is not restricted for event trigger detection, and can conduct the detection and classification of event triggers simultaneously, thereby avoiding the propagation of errors.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we presented a method for identifying an event trigger for wireless access to a flash crowd prediction based on a recurrent neural network GRU, which is shown to be capable of recognizing triggers and classifying them into correct event types. First, we proposed a text encoding method that allows deep learning application techniques to be applied. This work can function as a reference for event detection, event extraction, event prediction, and so on. Second, we proposed a GRU-based trigger identification and classification to automatically learn robust representations of event triggers without expert knowledge. As a result of the trigger identification and classification validation, our approach achieves a good performance in terms of accuracy, precision, recall, and F1 value.

Furthermore, our comparative experiments verified that a better performance can be achieved when lexicon and sentence features are encoded into an input vector. As one limitation of our approach, a word segmentation error is difficult to avoid for error propagation; in addition, the lack of a sufficient corpus is another limitation that needs to be tackled.

In a future work, we plan to study and design more general encoding methods that can mitigate or even avoid using a word segmentation; we will try to explore a domain-adaptation transfer learning based method to solve the so-called small-sample-size (3S) problem, making it possible to achieve a better performance with a small corpus.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable suggestions, which helped us considerably in enhancing this paper.

REFERENCES

[1] R. Ameera, R. M. Rollya, and J. Jacob, "Dynamic base station or relay station deployment and small cell on/off strategy for green ultra dense networks," *Procedia Comput. Sci.*, vol. 143, pp. 827–834, Jan. 2018.

[2] X. Liu, X. Li, X. Chen, Z. Liu, and S. Li, "Location correction technique based on mobile communication base station for earthquake population heat map," *Geodesy Geodyn.*, vol. 9, no. 5, pp. 388–397, 2018.

[3] Y. Han, X. Xu, and Y. Li, "Multi-document automatic summarization of network news based on event extraction," *Chin. J. Inf. Sci.*, vol. 26, no. 1, pp. 58–67, 2012.

[4] D. Ahn, "The stages of event extraction," in *Proc. Workshop Annotating Reasoning Time Events*, Jul. 2006, pp. 1–8.

[5] H. Llorens, E. Saquete, and B. Navarro-Colorado, "TimeML events recognition and classification: Learning CRF models with semantic roles," in *Proc. 23rd Int. Conf. Comput. Linguistics*, Beijing, China, Aug. 2010, pp. 725–733.

[6] Y.-Y. Zhao, B. Qin, and W.-X. Che, and T. Liu, "Research on chinese event extraction," *J. Chin. Inf. Process.*, vol. 22, no. 1, pp. 3–8, 2008.

[7] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, Jul. 2015, pp. 167–176.

[8] T. H. Nguyen and K. G. R. Cho, "Joint event extraction via recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, San Diego, CA, USA, Jun. 2016, pp. 300–309.

[9] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, Sofia, Bulgaria, Aug. 2013, pp. 73–82.

[10] A. V. U. P. Kumar, A. M. Reddy V, and D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks," in *Proc. 3rd Int. Workshop Middleware Pervasive Ad-Hoc Comput.*, Dec. 2005, pp. 1–8.

[11] X. Feng, L. Huang, D. Tang, B. Qin, H. Ji, and T. Liu, "A language-independent neural network for event detection," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 66–71.

[12] Y. Zeng, Y. Yang, Y. Feng, Z. Wang, and D. Zhao, "A convolution BiLSTM neural network model for Chinese event extraction," in *Proc. 5th Int. Conf. Comput. Process. Oriental Lang.* Kunming, China: Springer, 2016, pp. 275–287.

[13] Z. Zhang, "Research on event extraction based on structured learning," M.S. thesis, Beijing Univ. Posts Telecommun., Beijing, China, 2017.

[14] H. Wang and H. Qi, "Event research on combination of trigger expansion, neural network and dependency analysis," *Softw. Guide*, vol. 17, no. 1, pp. 19–21, 2018.

[15] T. Mikolov, K. Chen, and G. Corrado, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781

[16] *GitHub—fxsjy/jieba: Stuttering Chinese Participle [EB/OL]*. Accessed: Mar. 15, 2018. [Online]. Available: https://github.com/fxsjy/jieba

[17] *GitHub—HIT-SCIR/ltp: Language Technology Platform [EB/OL]*. Accessed: Mar. 15, 2018. [Online]. Available: https://github.com/HIT-SCIR/ltp

[18] Z. Zhong, "Event ontology and its application in query expansion," Ph.D. dissertation, Shanghai Univ., Shanghai, China, 2011.

[19] J. Fu *et al.*, "Event recognition based on dependency analysis," *Comput. Sci.*, vol. 36, no. 11, pp. 217–219, 2009.

[20] Y. Hao, Q. Zheng, and Y. Chen, "Anomalous behavior recognition for network public opinion data," *J. Comput. Res. Develop.*, vol. 53, no. 3, pp. 611–620, 2016.

[21] Z. Chen and H. Ji, "Language specific issue and feature exploration in Chinese event extraction," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Boulder, CO, USA, Jun. 2009, pp. 209–212.

[22] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, Jul. 2015, pp. 365–371.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[24] K. Cho, M. B. Van, and C. Gulcehre, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[25] A. Farzindar and W. Khreich, "A survey of techniques for event detection in Twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, 2015.

[26] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1180–1197, May 2017.

[27] X. Xuan, T. Liao, and B. Gao, "Research on automatic extraction of Chinese event triggers," *Comput. Digit. Eng.*, vol. 43, no. 3, pp. 457–461, 2015.

[28] B. Ma, "Research on key technologies of event relationship identification," M.S. thesis, Soochow Univ., Suzhou, China, 2014.

[29] F. Chollet. *Keras Github Repository.[EB/OL]*. Accessed: Nov. 15, 2018. [Online]. Available: https:github.com/fchollet/keras

[30] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. Workshop New Challenges NLP Frameworks*, 2010, pp. 45–50.

[31] D. Chen, N. Zhang, N. Cheng, K. Zhang, Z. Qin, and X. S. Shen, "Physical layer based message authentication with secure channel codes," *IEEE Trans. Depend. Sec. Comput.*, to be published. doi: 10.1109/TDSC.2018.2846258.

[32] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu, and X. Shen, "Synergy of big data and 5G wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 12–18, Feb. 2018.

[33] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 16–22, Aug. 2009.

[34] D. Chen, N. Zhang, R. Lu, N. Cheng, K. Zhang, and Z. Qin, "Channel precoding based message authentication in wireless networks: Challenges and solutions," *IEEE Netw.*, vol. 33, no. 1, pp. 99–105, Jan./Feb. 2019.

[35] D. Chen, N. Zhang, R. Lu, X. Fang, K. Zhang, Z. Qin, and X. Shen, "An LDPC code based physical layer message authentication scheme with prefect security," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 748–761, Apr. 2018. doi: 10.1109/JSAC.2018.2825079.

[36] Q. Wang, D. Chen, N. Zhang, Z. Qin, and Z. Qin, "LACS: A lightweight label-based access control scheme in IoT-based 5G caching context," *IEEE Access*, vol. 5, pp. 4018–4027, 2017.

[37] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, and X.-Y. Li, "S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 88–100, Feb. 2017.

MINGYONG YIN received the B.S. degree from Fudan University, in 2006, and the M.S. degree from the China Academy of Engineering Physics, in 2008. He is currently pursuing the Ph.D. degree with the College of Computer Science, Sichuan University. He is a Senior Engineer of the Institute of Computer Application, China Academy of Engineering Physics. His research interests are both theoretical and experimental, including natural language processing, cyberspace security defense and advanced threat detection method, threat intelligence auto collection and generation, and social network propagation dynamics.

XINGSHU CHEN received the Ph.D. degree from Sichuan University, in 2004, where she is currently a Professor with the College of Computer Science and the Cybersecurity Research Institute. She is currently a member of the China Information Security Standardization Technical Committee. Her research interests include cloud computing, cloud security, distributed file systems, big data processing, network protocol analysis, and new media supervision.

HAIZHOU WANG received the B.E. and Ph.D. degrees from the College of Computer Science, Sichuan University, China, in 2008 and 2014, respectively. From 2013 to 2014, he visited the Department of Electrical and Computer Engineering, University of Toronto, under the supervision of Prof. J. Liebeher. Since 2018, he has been an Associate Professor with the College of Cybersecurity, Sichuan University, China. His research interests include peer-to-peer IPTV systems, information security, network measurement, public opinion monitor, and intelligence analysis.

QIXU WANG received the B.Sc. degree from the Southwest University of Science and Technology, in 2009, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China, in 2017. He is currently an Assistant Researcher with the College of Cybersecurity, Sichuan University. His current research interests include cloud computing security, wireless network security, trusted computing, and data privacy protection.

CHENXI MA received the B.E. and M.S. degrees from the College of Computer Science, Sichuan University, China, in 2015 and 2018, respectively. Her research interests include natural language processing, information extraction, sentiment analysis, and social networking.

XUE QIN received the B.S. degree in telecommunication engineering from the North University of China, in 2008, and the master's degree from the Conestoga College, Canada, in 2017. She is currently pursuing the degree with the Department of Computing Sciences, Texas A&M University at Corpus Christi, TX, USA. Her research interests include communication networks, security, and machine learning.