Event Detection from Web Data in Chinese Based on Bi-LSTM with Attention

Yuxin Wu[1], Zenghui Xu[2], Hongzhou Li[3](✉), Yuquan Gan[4], Josh Jia-Ching Ying[5], Ting Yu[2], and Ji Zhang[6](✉)

[1] School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

[2] Zhejiang Lab, Hangzhou, China
{xuzenghui,yuting}@zhejianglab.com

[3] School of Life and Environmental Sciences, Guilin University of Electronic Technology, Guilin, China
homzh@163.com

[4] School of Telecommunication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China
ganyuquan@xupt.edu.cn

[5] Department of Management Information Systems, National Chung Hsing University, Taichung 402, Taiwan ROC

[6] School of Mathematics, Physics and Computing, The University of Southern Queensland, Toowoomba, Australia
Ji.Zhang@usq.edu.au

Abstract. Events are important activities people are involved in real life, and the information about events may be fascinating and important for people to understand and keep abreast with the key developments of some important social and individual subjects. In the big data era, event detection methods can help people efficiently and quickly extract specific information from massive Web information. However, the existing methods usually load the entire Web page information as the input into the models, and the rich noise and irrelevant information on Web pages will seriously impact the event detection performance of these methods. Also, the existing methods mostly used static models, which fail to consider the dynamics of information on the Web. To improve the performance of event detection and classification, we propose in this paper a new method that partitions the Web pages into multiple text blocks and utilizes Bi-LSTM with the attention mechanism for fine-grained event detection from Chinese Web pages. We also propose a dynamic method that updates the data as well as the model regularly and incrementally, making our model more adaptive to the ongoing changes of the Webpage data. The experimental results show that our model outperforms existing methods in event detection in terms of detection performance, the associated computational overhead, and the ability to deal with evolving Webpage information.

Keywords: Event detection · Text blocks · Dynamic maintenance

This research is supported by the Natural Science Foundation of China (No. 62172372), Zhejiang Provincial Natural Science Foundation (No. LZ21F030001) and Zhejiang Lab (No. 2022KG0AN01).

W. Chen et al. (Eds.): ADMA 2022, LNAI 13725, pp. 93–105, 2022.
https://doi.org/10.1007/978-3-031-22064-7_8

1 Introduction

In our daily lives, people are involved in various different types of events that occur at specific times and locations, which are important information for people to understand the latest developments of some important social and individual subjects. Examples of such events in the context of academic organizations, such as universities, could be graduation ceremonies, academic visits and research seminars, etc. Events are an important carrier of information that help us recognize and understand the real world [15].

As more and more information is digitized and published online (e.g., as in Webpages), it becomes possible to collect a large amount of information regarding different types of events in order to produce a collection of events that are interesting to different cohorts of users. However, in the big data era, effectively and efficiently generating such event-based information is challenging. Firstly, there is a large number of potential Webpages, even for a single organization, that potentially contain event-based information. Secondly, too much irrelevant information makes it nontrivial to accurately obtain information about specific events, which are treated as noise and may impact event detection in an adverse manner. Last but not least, the information on Webpages are dynamic in the sense that most of them are constantly evolving. New Webpages may be created that carry event-based information that should be periodically collected and analyzed to detect events for users. The above challenges will make event detection methods time-consuming, inaccurate and lack of the ability to deal with evolving event information. Therefore, how to quickly, accurately, and dynamically detect and classify event information from massive and noisy Webpages is the key problem to be solved in this paper.

So far, researchers have developed and proposed in literature different event detection, and classification methods, which use IT and intelligence technologies to automatically detect and classify events from massive data [21]. Obviously, such automatic event detection and classification technologies can alleviate the problems caused by information overloading in the big data era and improve the efficiency of people's acquisition of event-related information. However, these methods suffer some key drawbacks as follows:

a) The methods based on pattern matching rely on specific linguistic knowledge and requires domain knowledge of experts, also the process of pattern formulation is time-consuming, laborious, and prone to errors. The method based on machine learning is limited in the extraction of deep text features. The relatively simple and shallow neural networks are prone to convergence to the local minimum, which affects event detection performance.
b) The methods based on deep learning usually take the whole text content of Webpages as the input corpus in the model training stage, which includes a large amount of irrelevant information. Therefore these methods suffer from inferior detection performance.
c) Most event detection models are static and lack the mechanism of continuous maintenance. These models stops iteratively updating the detection model

after training is completed or are not able to incrementally update the model when new Webpages are generated. They can always generate new models from scratch, but it is apparently too computationally expensive and not efficient at all when dealing with large evolving information on Internet.

To address aforementioned drawbacks, this paper propose a Web-based events deep detection model(Attention with Bi-directional Long Short-Term Memory, ABiLSTM). The incremental crawler technology is used to periodically obtain the latest updates of Webpages. The BiLSTM neural network is leveraged to learn effectively the text features of Webpages which are partitioned. The Attention mechanism is also introduced in the model to assign different weights to text to allow the model to focus on the key textual information indicative of the occurrence of events. Finally, the method of dynamic incremental data maintenance is adopted, which help achieve a good efficiency in dynamic detection of events without sacrificing the detection effectiveness. Specifically, the main contributions of this paper include:

1. The textual content of Webpages is partitioned into different blocks for a finer textual granularity, which can effectively minimize noisy information and capture the key textual content, considerably contributes to the better event detection performance.
2. To effectively capture and model the textual information for event detection, the BiLSTM neural network with attention mechanism is leveraged in our model to learn effectively the text features of partitioned Webpages.
3. Our model supports the dynamic maintenance of the event detection results. By periodically retraining the model with newly incremental Webpages information, our model can iterate continuously.
4. The experimental results show that our model outperforms existing methods in event detection in terms of classification effectiveness and the ability to deal with evolving Webpage information.

2 Related Work

2.1 Pattern Matching Based Methods

Pattern matching based methods manually design the feature template of the corresponding event in advance, and then match it with the target text.

Zhihu et al. proposed an event-adaptive concept integration algorithm, which estimates the effectiveness of semantically related concepts by assigning different weights, and realizes the use of related concepts to match a target event differently and achieve better results [8]. Song Qing et al. identified key sentences in news articles according to word frequency and text clustering and performed subsequent identification and extraction of key sentences [13].

Although the structure of the event detection methods based on pattern matching is intuitive, similar to the way of human thinking, and easy for people to understand, these methods rely on the knowledge of specific domain experts, the time and labor costs are high in the process of pattern formulation, and errors are prone to occur. At the same time, the specific pattern is only for specific fields, the generalization ability of the model is generally poor.

2.2 Machine Learning Based Methods

Machine learning based methods mainly focuses on the discovery of features and the construction of classifiers, which transform event detection into classification.

Traditional Machine Learning Methods. In the past, the text classification mostly used the bag-of-words model, N-grams and its word frequency-inverse document frequency as features, and the major traditional models for text classification include naive Bayes [3], support vector machines, k-nearest neighbors, and hidden Markov models. Chieu and Ng first introduced the maximum entropy classifier into event extraction and used the classifier to detect events and their elements [2]. Zhang et al. manually designed features for Chinese news texts and used CRF to identify and extract trigger words [20]. However, the traditional machine methods, including shallow neural networks, in most cases are not very effective in modeling the textual information of Webpages for the purpose of event detection and classification particularly given the presence of ample noises in the Webpages involved.

Deep Learning Based Methods. Nowadays, with the rise of deep learning and the more efficient extraction of text features by word embedding methods, the neural network model in deep learning has been more and more applied in the field of NLP. The word vector representation model word2vec proposed by Mikolov in 2013 has achieved good text representation results [12]. Kim first used a CNN network model in 2014 to deal with the classification of textual data, and the method performed well in multiple text classification tasks [6]. Lai and niu proposed a model RCNN based on RNN concatenated CNN and achieved good results in classification [10]. Zhou et al. took advantage of the selective retention of long-sequence information by memory cells in the LSTM to extract the sequence information of the text to achieve text classification [22]. Zhang et al. compared the impact of word embeddings at different dimensions on the results of text classification [20]. Lin et al. added a self-attention mechanism to the sentence classification problem to deeply mine the internal relations between sentences and text context semantics, so as to improve the classification performance of the model [9]. The Bert model proposed by Devlin in 2018 is a two-way language model in a complete sense, which can take into account the text sequence information, context information, and grammatical context information in the whole sentence, avoiding the problem of polysemy [4]. Since then, classification based on Bert model has been applied in many fields, such as CM-BERT [18] and BERT4TC [19].

It can be seen from the above research work that, no matter whether it is based on pattern matching or machine learning event detection methods, firstly the original data are event Webpages, secondly, it is necessary to obtain relevant event features, such as trigger words [5,17], entities [7,14], dependencies [1], etc., to detect events. However, an overwhelmingly majority of Webpages for a given Website may typically be non-event Webpages, and it is necessary to detect event-type texts before classifying them as events. The non-event-type data has no clear theme, the text format is not fixed and various. Moreover, it is usually that a few key sentences, rather than the whole Web-page, indicate that this is a Webpage to represent an event. Existing work also confirmed that traditional methods of inputting the whole training corpus into the model have a poor performance, while a model with attention mechanism can be better. Therefore, we adopt in this paper the idea of text partition of Webpages to reduce the textual granularity for training our detection model in a fine-grained manners.

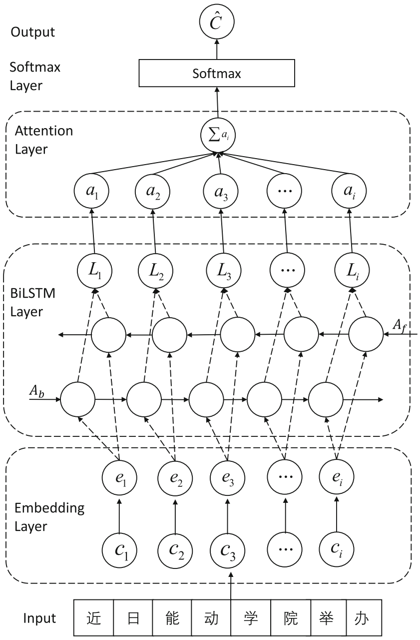

Fig. 1. Structure diagram of our model

3 ABiLSTM Model

3.1 Problem Formulation

Given a Webpage $C = (c_1, c_2, \ldots, c_i)$, classify the Webpage into a binary category in $S = (s_1, s_2)$, and then if C is an event Webpage, we further multi classify it into a category in $R = (r_1, r_2, r_3, ..., r_j)$. Among them, c_i is the word set of each Webpage, s_1, s_2 corresponds to event and non-event Webpages, respectively, and $r_1, r_2, ..., r_j$ correspond different categories of events. For example, these categories can be student life, scientific research, work updates and others in a university Website.

3.2 Static Classification Model

Figure 1 presents the overall structural architecture of our model. Before the data is loaded into model for both training and testing, the Webpage text is divided into blocks. Specifically, our model contains the following several layers:

Input and embedding layer. For the input long text $C = (c_1, c_2, \ldots, c_i)$, first map each word to the word vector of the corresponding dimension through Word2vec, and then multiply it with the weight matrix obtained by One-hot encoding, and finally get each word embedding to represent e_i.

Bi-LSTM layer. The feature vector e_i is input to the forward loop unit A_b and the reverse loop unit A_f of the LSTM, and the forward output $\overrightarrow{h_i}$ and the reverse output $\overleftarrow{h_i}$ are obtained. Finally, the output L_i of this layer is spliced by $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$. The formulas involved in this step are:

$$f_t = \delta\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$i_t = \delta\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \delta\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

where i_i, f_t, and o_t are the input gate, forget gate and output gate respectively; C_{t-1}, $\tilde{C}_t$, C_t are the memory unit value at the previous moment, the candidate memory unit value at the current moment, and the memory unit state value at the current moment respectively; W and b represent the trainable parameters;

Attention layer. Then in the attention layer, we calculate the similarity between the query vector Q and the key vector K_i of each data feature. The *softmax* function is used to normalize the attention score to obtain the weight distribution. According to the weight coefficient, the Value is weighted and summed, and the calculation process of the final feature at time t is shown in formula (7):

$$\text{Attention } = \sum_{i=1}^{N} \mathrm{a}_i \cdot V = \sum_{i=1}^{N} \frac{\exp\left(\mathrm{h}_t\right)}{\sum_{j=1}^{N} \mathrm{h}_j} \cdot V \tag{7}$$

where i represents the number of input text sequences, h_t is the hidden layer state sequence at time t, h_j is the hidden vector corresponding to the j feature word in the text at time t; a_i is the weight of the hidden layer state h_t at time t to h_j distributed;

Softmax layer. The output data is passed to the *softmax* layer to be transformed into a probability output, and the label value is output through the *argmax* function.

3.3 Dynamic Model Maintenance

In the era of big data, Web pages frequently update event information, and the classified Web pages may have content changes or generate a large number of new Web pages. To adapt to the dynamics of network events, ensure the effectiveness of the model and prolong the life cycle of the model, this paper adopts the strategy of dynamic maintenance for the classification model. This paper proposes to use an incremental crawler to crawl Web pages regularly, use MD5 value to quickly check whether the content of Web pages has changed from the latest time snapshot, filter duplicate Web pages, and collect new data incrementally. Compared with traditional maintenance from scratch, dynamic maintenance can effectively reduce the data collection time and model training time, making the method in this paper more lightweight.

The dynamic maintenance strategies are explored in our work. Specifically, at a certain time node t_2, we use the time period from the previous time node t_1 to t_2 to generate new text data or text data with content changes, load them into the m_1 model that has been generated on the t_1 time node, and finally output the model m_2 on the t_2 time node. The specific algorithm flow is presented in Algorithm 1.

4 Experimental Evaluation

4.1 Dataset and Experimental Setup

The dataset of this paper comes from the texts on all Web pages of Nanjing University of Aeronautics and Astronautics (http://www.nuaa.edu.cn). Comprehensively crawl [11,16]the Website through the crawler tool, take NUAA news Web page and NUAA Graduate School Web page as the starting URL, and set the list of allowed crawling domain names to ensure that the crawlers are all running under a specific domain name (nuaa.edu.cn), filtering irrelevant URLs. During the crawling process, the crawler tool will first compare the MD5 value of the current Web page with the existing MD5 value in the database. If there is no repetition, grab all text content on the Web page, including navigation bar, title, text, time, and other features, and finally save it in TXT file format with the file name MD5 value; If there are duplicates, skip the current Web page.

Dataset 1 of the two classification problems in this paper has a total of 7956 text data, including 2135 event-type text data and 5821 non-event-type text data. Dataset 1 is divided at a ratio of 7:3, of which 70% of the data is used as the training set, with a total of 5569 records; 30% of the data is used as the test set, with a total of 2387 records. Event-type text data, such as school news, is marked as 1; non-event-type text data such as personal introduction, navigation bar, rules, legal text, is marked as 0.

Dataset 2 of the multi-classification problem in this paper has a total of 8618 text data, which can be divided into 5 categories according to the category dimension, including 1761 text data for campus life, 1232 text data for academic research, and 3215 text data for communication and visit. 2,105 pieces of work dynamic text data and 305 pieces of other text data; According to the time dimension, it can be divided into 4-time nodes, and the interval between each time node is 60 days. At the first time node t_1, 6678 pieces of data are collected, and at the second time node t_2, 1002 pieces of data are added. 565 pieces of data are added at the third time node t_3, and 373 pieces of data are added at the fourth time node t_4. Category 1 events in dataset 2 are for students campus life, focusing on students'daily life; Category 2 events are for scientific research, focusing on academic exchanges between teachers in various colleges in the school; Category 3 events are for school work, focusing on relevant work in the school; Category 4 events are for research cooperation and exchange, focusing on exchanges with organizations outside the University; Category 5 events are for others that do not belong to any of the aforementioned four categories.

The experiments are performed on Windows 10 professional platform, using Python 3 as the compiler language and Pycharm community integrated development environment.

4.2 Chinese Text Preprocessing

First, manually label dataset 1 and dataset 2, so that the data corresponds to the label one by one. Next, the data is cleaned to remove invalid characters, punctuation marks, etc. Then, the long text in dataset 2 is divided into several blocks, the size of each text block is adjusted by setting the number of words in the block, and finally, Chinese word segmentation is realized by the Jieba word segmentation tool.

4.3 Sensitivity Analysis

We carry out the sensitivity study of our proposed model and several mainstream baseline methods, including SVM, TextCNN, Bi-LSTM,Bert, in terms of precision rate(P), recall rate(R) and F1-score as the evaluation metrics.

We study the effectiveness of all the methods under varying block sizes, denoted by k. In the dataset 2 experiments, the block size, k, is varied from 25 words to 100 words. Table 1 presents the F1-Score of different methods under different block sizes. The table shows that different block sizes have an impact on the classification results, with k = 75 words being the most appropriate value for our model. The block size is 25 words, the text block does not contain the complete event information to properly train the model; when the size is 100 words or above, the noises in the text block increases, which adversely affects the model learning.

Table 1. The F1-Score of the algorithm under different block sizes

Models	k = 25 words	k = 50 words	k = 75 words	k = 100 words
SVM	77.6%	81.5%	84.5%	83.2%
TextCNN	79.6%	83.1%	88.6%	85.1%
BiLSTM	80.2%	85.6%	89.2%	86.8%
Our model	**81.4%**	**86.8%**	**90.3%**	**88.2%**

Algorithm 1. Dynamic Maintenance Model Algorithm Process

Input: New long text data or changed text data generated between t_1 and t_2 time nodes $C = (c_1, c_2, \ldots, c_i)$,$c_i(i \in [1.i])$ is a single Chinese word;

Output: Event class label R;

1: Load the ABiLSTM model m_1 output at time t_1;

2: Input the word c_i into the Embedding layer and convert it into the embedded word $e_i = W^{wrd}v^i$, $W^{\mathrm{wrd}} \in R^{d^w|V|}$ The output is the embedded word vector of the long text $emb_s = \{e_1, e_2, \ldots, e_n\}$;

3: The text embedding vector $emb_s = \{e_1, e_2, \ldots, e_n\}$ is used as the input of the m_1 model, and the output is:

$$L_i = \mathrm{LSTM}\left(e_i\right), i \in [1, n]$$

4: Input the L_i generated by the previous layer to the Attention layer, and obtain implicit information through nonlinear transformation, and the output is:

$$a_i = \sum \alpha_i L_i = \sum \frac{\exp\left(\boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{h}_w\right)}{\sum_t \exp\left(\boldsymbol{h}_i^{\mathrm{T}} \boldsymbol{h}_w\right)} L_i$$

5: Use a_i as the input of the softmax layer to get the probability of the corresponding category of the current input long text:

$$R = \mathrm{argmax}\,\mathrm{softm\hat{a}x}\left(a_i\right)$$

6: Output ABiLSTM model m_2 at time t_2

7: **return** R

4.4 Effectiveness Analysis

In this section, we carried out two experiment that compares the performance of models based on dataset 1 and dataset 2.

Table 2. Models performance based on dataset 1

Models	Event-type	Non-event-type
SVM	86.9%	82.5%
TextCNN	94.3%	93.2%
LSTM	95.1%	94.2%
Our model	96.2%	97.2%

Compared with non-event-type data, the event-type data of dataset 1 has obvious keyword characteristics and certain event elements, at the same time the boundary between the two is clear, and the overlapping part is less. As can be seen from the Table 2, the baseline models have achieved good classification results, especially the F1 value of our model for event-type data is 96.2%.

As can be seen from the Table 3, the classification effect of deep learning model is better than that of machine learning model. This is because the neural network can extract text features at a deeper level and learn more key information. The RNN neural network is better than CNN because the BiLSTM unit

Table 3. Models performance based on dataset 2

Models	Mic_Precision	Mic_Recall	Mic_F1-score	s/epoch
SVM	0.763	0.778	0.77	–
TextCNN	0.821	0.855	0.838	58
LSTM	0.845	0.875	0.86	78
BiLSTM	0.871	0.902	0.886	108
Bert	**0.902**	**0.932**	**0.917**	**361**
Our model	**0.887**	**0.921**	**0.893**	**134**

can retain or discard the current content features according to their importance on the one hand, and effectively extract the connection between the long-term sequences in this article; on the other hand, from the two Read the sequence data in each direction, capture the information of the left and right contexts, connect and summarize to complete the prediction, and extract the semantic information of the text at a deeper level. The introduction of the attention mechanism has helped the model to be slightly improved. The role of keywords is highlighted by assigning weights, and the model is more targeted for learning.

Finally, the classification effect of the model in this paper is slightly behind the BERT model, because Bert is a pre-training model, and the scale of the data set used in its training is hundreds to thousands of times that of this paper. A larger data set can enable deeper information on text features. The capture of semantic information is closer to the reading comprehension of the human brain; next, a transformer with very good parallel performance and a deeper and wider scale can be used in the selection of encoder, so that words at any position can ignore the restrictions caused by direction and distance, and have the opportunity to integrate each word directly. Although Bert is an excellent pre-training model, and the classification effect of the model is better, even its fine-tuning consumes huge computational resources. The Bert model takes 361s to train an epoch, while the ABiLSTM model takes 134s, which is only 1/3 of that of Bert when the difference in classification effect is not very large. Therefore, based on the experimental results, this paper selects the ABiLSTM model to complete the multi-classification problem in this paper.

4.5 Dynamic Maintenance Comparison

In this section, we compared the performance of the two model maintenance strategies, the incremental maintenance and the maintenance from scratch.

Table 4. The F1-Score of the algorithm under the two maintenance methods

Maintenance mode	t_1	t_2	t_3	t_4
Incremental maintenance	0.823	0.855	0.87	0.879
Maintenance from scratch	0.834	0.874	0.887	0.893

Table 5. The cost of computing resources under the two maintenance methods

Maintenance mode		$0\tilde{}t_1$	$t_1\tilde{}t_2$	$t_2\tilde{}t_3$	$t_3\tilde{}t_4$
Incremental maintenance	h/data collection time	26	3	2	2
	s/epoch	78	20	20	16
Maintenance from scratch	h/data collection time	26	30	33	35.5
	s/epoch	78	105	120	138

It can be seen from Table 4 that the model classification effect of maintenance from scratch is better than that of incremental maintenance. This is because the amount of data input to the model is different between the two methods except at time t_1. For example, at the t_2 time node, 7680 pieces of data are input to the model maintenance from scratch, while only 1002 pieces of data are provided by incremental maintenance. But what contrasts sharply with the less obvious classification effect is the time spent on data collection for both approaches. Maintenance from scratch, when crawling at the beginning, while searching for URLs and saving texts, it often takes more than one day to obtain a complete data set of the current time node. As shown in the Table 5 incremental maintenance takes a lot of time to crawl for the first time. For the rest of time, the crawler will compare the MD5 value first, and crawl if there is no repetition. The number of newly generated Web pages per unit time interval is not large, so it takes less time. Considering the computational resource overhead and classification effect, this paper maintains data in a dynamic incremental manner.

5 Conclusion and Future Work

In this paper, a Bi-LSTM deep learning model with the attention mechanism is proposed for event detection. The Webpage textual content is partitioned into blocks with a fixed size, whereby the key information contributing to event detection is extracted for training our model in order to achieve better detection accuracy. Incremental model maintenance strategies are applied to continuously maintain our model using incrementally crawled Webpages information to achieve efficient model maintenance.

In the future, we will carry out more comprehensive experiments by creating additional real-life datasets from organizations other than universities. An interactive system will be developed for presenting the detected events in a more visualized and interactive manner across different platforms. We will also extend the model to deal with other languages, such as English, where such adaption, we believe, is straightforward.

References

1. Can, E.F., Manmatha, R.: Modeling concept dependencies for event detection. In: Proceedings of International Conference on Multimedia Retrieval, pp. 289–296 (2014)

2. Chieu, H.L., Ng, H.T.: A maximum entropy approach to information extraction from semi-structured and free text. Aaai/iaai **2002**, 786–791 (2002)
3. Cui, W.: A Chinese text classification system based on naive bayes algorithm. In: MATEC Web of Conferences, vol. 44, p. 01015. EDP Sciences (2016)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Ding, N., Li, Z., Liu, Z., Zheng, H., Lin, Z.: Event detection with trigger-aware lattice neural network. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 347–356 (2019)
6. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
7. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 297–304 (2004)
8. Li, Z., Yao, L., Chang, X., Zhan, K., Sun, J., Zhang, H.: Zero-shot event detection via event-adaptive concept relevance mining. Pattern Recogn. **88**, 595–603 (2019)
9. Lin, Z., et al.: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017)
10. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)
11. Lu, H., Zhan, D., Zhou, L., He, D.: An improved focused crawler: using web page classification and link priority evaluation. Math. Probl. Eng. **2016**, 1–10 (2016)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Qing, S., Ying, Z., Pengzhou, Z.: Research review on key techniques of topic-based news elements extraction. In: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), pp. 585–590. IEEE (2017)
14. Sayyadi, H., Hurst, M., Maykov, A.: Event detection and tracking in social streams. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 3, pp. 311–314 (2009)
15. Sims, M., Park, J.H., Bamman, D.: Literary event detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3623–3634 (2019)
16. Singh, B., Gupta, D.K., Singh, R.M.: Improved architecture of focused crawler on the basis of content and link analysis. Int. J. Mod. Educ. Comput. Sci. **9**(11), 33 (2017)
17. Tong, M., et al.: Improving event detection via open-domain trigger knowledge. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5887–5897 (2020)
18. Yang, K., Xu, H., Gao, K.: CM-BERT: cross-modal BERT for text-audio sentiment analysis. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 521–528 (2020)
19. Yu, S., Su, J., Luo, D.: Improving BERT-based text classification with auxiliary sentence and domain knowledge. IEEE Access **7**, 176600–176612 (2019)
20. Zhang, C., Hong, S., Zhang, P.: The research on event extraction of Chinese news based on subject elements. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–5. IEEE (2016)

21. Zhong, Z., Jin, L., Feng, Z.: Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 96–100. IEEE (2015)
22. Zhou, C., Sun, C., Liu, Z., Lau, F.: A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630 (2015)