Hybrid Syntactic Graph Convolutional Networks for Chinese Event Detection

Xiaobo Ma[1,2], Yongbin Liu[1,2(✉)], and Chunping Ouyang[1,2]

[1] University of South China, Hunan, China
[2] Hunan Medical Big Data International Sci.&Tech. Innovation Cooperation Base, Hunan, China

Abstract. Event Detection (ED) is a task that aims to recognize triggers and identify the event type in sentences. Syntactic information plays a crucial role in event detection model accurately to recognize the triggers and event type. The previous works commonly use word embedding to obtain context representation that cannot fully exploit syntactic information. In this paper, we propose a novel model HSGCN (Hybrid Syntactic Graph Convolutional Networks) for Chinese event detection, which utilizes graph convolutional networks to generate sentence-level feature and exploit a task-specific hybrid representation considering both character-level feature and word-level feature. Our experiments demonstrate that HSGCN model can capture rich syntactic to improve identifying the triggers and event type. Compared with existing models, HSGCN can achieve efficient and accurate results on ACE 2005 and KBPEval2017 datasets. In trigger identification and type classification tasks, HSGCN respectively achieved 70.2% and 65.7% F-score with average 1.2% and 0.9% absolute improvement compare to state-of-the-art method.

Keywords: Event detection · Graph convolutional networks · Syntactic information · Hybrid representation

1 Introduction

The content of events that defined by ACE is consisted of event triggers and event arguments. Event Detection (ED) is a key step in event extraction, which aims to recognize triggers and identify the event type. For example, in the sentence "Clashes between the police and protesters killed at least five people", an event detection system will detect a death event triggered by "killed". Event trigger is a single token that represent the occurrence of an event. A token is not equivalent to a word, for example, the phrase "step down" is concatenated into a token "step-down" as the trigger of a "personnel" event. This design has been used in numerous studied on ED.

Chinese Event Detection is more complex than English Event Detection, because triggers are not always exactly match with a word. In many cases, a trigger can be the part of a word or cross multiple words. What's more, some

H. Chen et al. (Eds.): CCKS 2020, CCIS 1356, pp. 147–159, 2021.
https://doi.org/10.1007/978-981-16-1964-9_12

related words that we call cue words in the sentence can provide available information to assist the trigger classification. These cue words are the crucial factor to identify event type. However, sometimes an event may has more than one cue word, and different cue word is located in the different parts of sentence. Therefore, traditional word embedding is difficult to consider these cue words because they are scattered and far away from the triggers. To solve this problem, we use dependency tree to link cue words to the trigger words, which can connect a dependency word to a head word and provide their dependency relation.

Some previous works like [21,29] had employed syntactic dependency information to generate syntactic representation for event factuality prediction. Usually, these models generated a syntactic representation by one-hot encoding and concatenated syntactic representation to context representation in the final stage. Although integrating syntactic information into representations had been proved to be effective in ED task, using the simple combination would cause noise data appearing in syntactic and context representations, even lost some useful information. In order to address these problems, we introduce graph convolutional networks [13,17,20] to represent syntactic information flexibly, without requiring special encoding for dependency information as in previous works.

Compare with English, Chinese has a particularity that a word can be composed of multiple characters and these characters have their own meanings. For example, in Fig. 1, the word "遭袭" (be attacked) composed of the character "遭" (encounter) and the character "袭" (attack). In this sentence, the word "遭袭" (be attacked) which is a whole trigger would be divided into two characters in dependency parser. Hence combination of word embedding and character embedding could keep the semantic of Chinese vocabulary effectively.

联合国/ 南苏丹/ 特派团/ 在/ 阿科博/ 的/ 营地/ 遭/ 袭/ 。

UN mission camp in South Sudan in Akobo attacked.

Fig. 1. An example of attack event which the trigger word is divided into two characters after tokening.

In view of the above discussion, a model called HSGCN (Hybrid Syntactic Graph Convolutional Networks) is proposed for Chinese event detection task. We conduct experiments on ACE 2005 and KBPEval2017 event extraction datasets. The experiment results demonstrate HSGCN outperforms previous models. The contributions of this paper include: (i) an important feature for triggers identification, syntactic dependency, is introduced to our model, and exploiting graph convolutional networks which is well-suited for handling dependency tree to represent syntactic dependency, (ii) considering the semantic characteristics of Chinese vocabulary, combining word-level feature and character-level feature could keep the original semantic of Chinese word to improve the triggers identification, and (iii) provides state-of-the-art performance on ACE 2005 and KBPEval2017 datasets in Chinese event detection.

2 Related Work

For event detection task, traditional classification models are mainly divided into two categories: feature engineering based model and machine learning based model. [4] presented a classification-based model towards single-slot and multi-slot information extraction which got a better performance than pattern-learning approaches. [1] employed maximum entropy to complete triggers detection and designed TiMBL classifier to distinguish event types. In follow-up studies, some researchers captured document-level features to improve ED model. [10] first exploited cross-document feature to break down the document boundaries. [9] used information about different types of events to make predictions, and designed a document-level prediction model to achieve effectively event extraction in document-level. Feature-based methods can realize a performance gain by enrich different forms of features. However, feature-based methods have a problem of heavily dependent on man-made, which is time consuming and error prone. To solve these problems, [25] learned features automatically by using machine learning model to combine linguistic and structural information. [12] used conditional random fields with carefully designed features for the task of identifying definition sentences.

As machine learning technology continues to improve, recently, researchers pay more attention on deep learning methods to accomplish event detection task. [11,19,24] introduced the non-consecutive convolution to traditional CNN models and achieved a great performance for event detection. [26] proposed a dynamic multi-pool convolutional neural network to encode sentence semantic clues on event extraction task. [22] enhance a recurrent neural network with dependency bridges to capture syntactic information for event extraction. [15] proposed three hybrid representation methods to combine character-level and word-level feature. [28] proposed a joint model to extract event mentions and entities by capture structural information. These models brought some valuable solutions to ED task. However, most of them were designed for English event detection task, which cannot be used directly in Chinese event detection task. Furthermore, the existing models take no account of complicated compositional structure and polysemy in Chinese, which resulted in performance degradation in Chinese event detection task. Inspired by the previous work [5,6,16,26], we design a hybrid syntactic graph convolutional networks (HSGCN) which represent syntactic dependency in sentence level based on GCN to improve the triggers identification. In additional, our method keeps as much as possible the original semantic of Chinese vocabulary by combining word-level feature and character-level feature, thus achieves better performance in Chinese event detection task.

3 HSGCN

Event detection generally can be regard as a multiclass classification problem. The target of event detection is recognizing trigger candidate and identifying the event type of trigger candidate. In Chinese event detection, a word can to be a trigger candidate or form a part of trigger candidate. Therefore, we use characters as basic units in our model to match the requirement of Chinese processing. It means that the classifier decide each character in the sentence whether forms a part of triggers. If so, the classifier would identify the position of the character in the triggers.

Our framework which is depicted in Fig. 2 consists of the following four components: (a) sentence encoding that represents the sentence in both word-level and character-level; (b) sentence-level feature extraction based on syntactic graph convolutional networks and lexical feature extraction; (c) hybrid representation learning by combining word embedding and character embedding; (d) trigger generator and event type classifier.

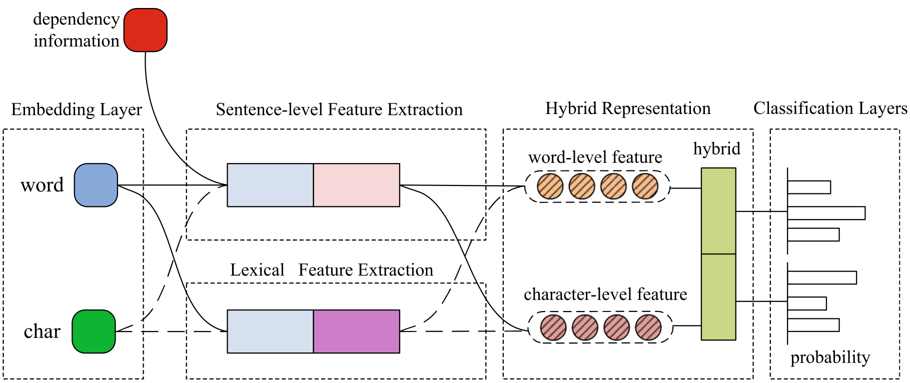

Fig. 2. The overall architecture of our model. The details of this architecture are depicted in Fig. 3, Fig. 4 and Fig. 6.

3.1 Sentence Encoding

Normally, the first step in event detection is to convert each word in the sentences into an embedding vector. In our work, we generate the final representation combining character-level representation and word-level representation. Therefore, the model encode sentences both in character-level and word-level.

We use the Skip-gram model [18] to obtain the character embedding and word embedding. Skip-gram model has been widely applied to many NLP tasks and achieve great performance. In particular, the sentence $S = (x_1, x_2, ..., x_n)$ would be fed in to Skip-gram model and trains the embedding of words or characters $x_1, x_2, ..., x_n$ by maximizing the average log probability,

$$\frac{1}{n}\sum_{t=1}^{n}\sum_{-w\leq i\leq w}\log p(x_{t+i}\mid w_t) \tag{1}$$

where w is the size of training window.

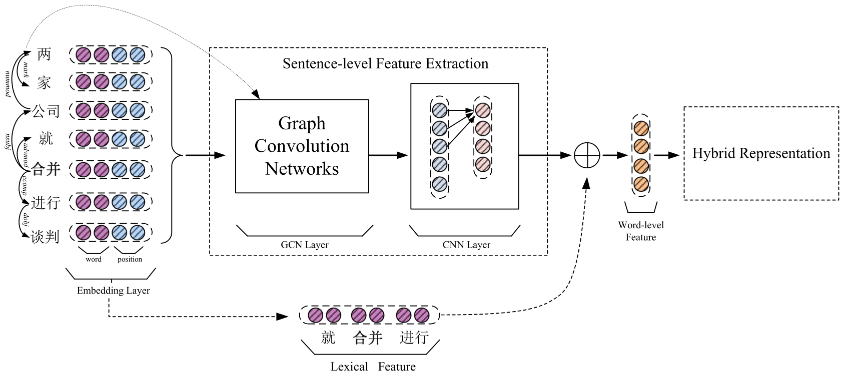

Fig. 3. The architecture that incorporating syntactic information in word-level feature In this example, the input is a sentence: 两家公司就合并进行谈判 (The two companies are negotiating a merger) and current word is "合并" (merger). ⊕ indicates a concatenation operation.

It should be noticed that the both character-level and word-level representation in our model is a lexical feature representation like [2] which can capture more rich semantics information than traditional word feature. Specifically, in Fig. 3, the current word is "合并" (merger) and its context tokens are "就" (as for) and "进行" (proceed). After generating three words embedding separately, our model concatenated the three words embedding into a whole word-level lexical feature to represent the word "合并". Similar to the case of word-level representation, in Fig. 4, the model also concatenated the character embedding of current character "并" and its context tokens into a whole character-level lexical feature.

3.2 Incorporating Syntactic Information to Sentence-Level Features

According to previous word [26], sentence-level clues can contribute significantly to event extraction. Concretely, in Fig. 5, a syntactic arc (along the *conj-arc* from conquered to attempted, along the *xcomp-arc* from attempted to invade) connect a trigger conquered to another trigger invade. From the trigger *conquered*, we obtain a clue to identify the event type about the trigger *invade*. Because of the clue word is far away from the candidate word, traditional method can't encode this dispersive context information. So the method considering sentence-level feature can effectively solve the above problem by learning the compositional semantic features of sentences. Taking full advantage of sentence-level feature,

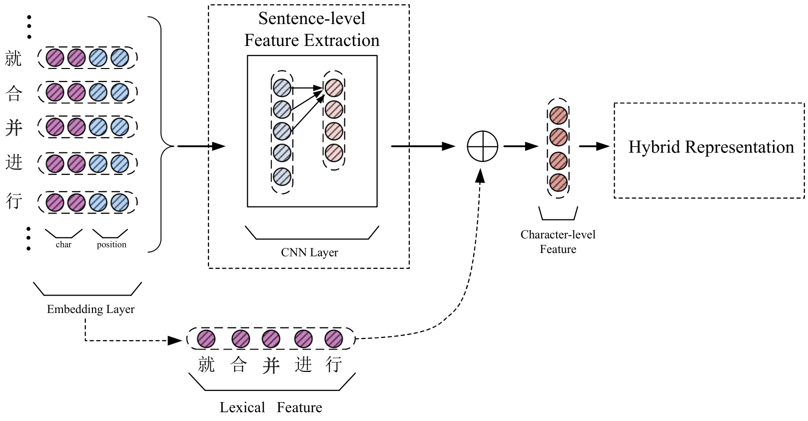

Fig. 4. The architecture to obtain character-level feature. $\oplus$ indicates a concatenation operation. The detailed architecture of hybrid representation is described in Fig. 6.

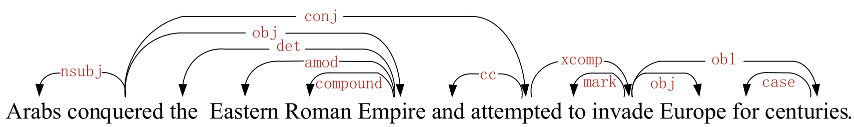

Fig. 5. The result of syntactic parser. The sentence contains two attack events respectively triggered by "conquered" and "invade".

we propose a novel feature extraction architecture to incorporate dependency information by using graph convolutional networks.

Let an directed graph $G = (V, E)$ be the dependency parsing tree for sentence $S = (x_1, x_2, ..., x_n)$, where V is the set of nodes ($|V| = n$) and E is the sets of edges. Each v_i in V indicates a token x_i. $(v_i, v_j, l_{ij}) \in E$ indicates the edge from node v_i to v_j with label l_{ij}. Due to the information of directed graph does not necessarily travel in its direction [13], we also add inverse edge (v_j, v_i, l_{ji}^{-1}) and all self-loops, i.e., (v_i, v_i). For the $k-th$ graph convolution network layer, model would calculate the vector $h_{v_j}^{(k+1)}$ of node v_j by:

$$h_{v_j}^{(k+1)} = f\Big(\sum_{v_i \in N(v_j)} (W_{l_{ij}}^k h_{v_i}^k + b_{l_{ij}}^k)\Big) \tag{2}$$

where $W_{l_{ij}}^k \in R^{d \times d}$ the weight matrix and $b_{lij}^k \in R^k$ is the bias for label. $N(v_j)$ indicates the set of neighbors of node v_j including v_j itself. In this work, we use Stanford CoreNLP toolkit [2] to generate the arcs in dependency parsing tree for sentences. It should be noticed that the first layer of graph convolutional networks is initialized by the output of embedding layer. As some of edges might be erroneous or irrelevant for the downstream task [23], we employ gates [17] on the edges to calculate their individual importance. For each edge (v_i, v_j), we calculate a score $g_{l_{ij}}^k$ indicating the importance for event detection by:

$$g_{l_{ij}}^k = \sigma(V_{l_{ij}}^k h_{v_i}^k + B_{ij}^k) \tag{3}$$

where $V_{l_{ij}}$ and B_{ij}^k are the weight matrix and bias of the gate, respectively, σ is the sigmoid function. Based the gating mechanism, the updated GCN propagation rulecan be formulated as

$$h_{v_j}^{(k+1)} = f(\sum_{v_i \in N(v_j)} g_{l_{ij}}^k \times (W_{l_{ij}}^k h_{v_i}^k + b_{l_{ij}}^k)) \tag{4}$$

In the proposed model, the GCNs are mainly be used to capture dependency information, while they can't capture sufficient information flow. In event detection, sequential context can be leveraged to expand information flow. Therefore, we feed output of GCN layer to a CNN layer and get the final sentence level feature.

3.3 Hybrid Representation Learning

As we shown in Fig. 3, the lexical feature and sentence-level feature make up the word-level feature. According to previous works [14,26], the word feature can provide more accurate semantic information, while the character feature can reveal the inner compositional structure of event triggers. Therefore, we use the similar architecture that described in Fig. 4 to obtain character-level feature. Then we incorporating word-level and character-level feature by a task-specific hybrid representation [15]. It should be noticed that only the word-level feature contain the dependency information in our case. In order to avoid excessively increasing the complexity of the model, we remove the GCN layer and the representation of dependency parser tree when we learn character-level feature.

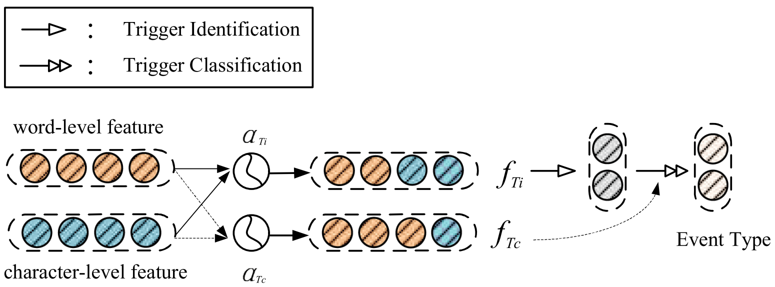

Fig. 6. Framework of event detection by hybrid representation learning. The word-level feature and character-level feature are respective the output of architecture in Fig. 3 and Fig. 4.

Chinese event detection can be divided into two processes: trigger identification and trigger classification. These two processes require different feature in a particular model. For example, trigger identification may rely on more structural information from character-level feature, while trigger classification need more semantic information from word-level feature. Consequently, two gates are trained to model the information flow for trigger identification and trigger classification. Figure 6 describes the framework of task-specific hybrid representation. We use $f^{'}_{word}$ and $f^{'}_{char}$ to separately represent word-level and character-level feature. The gates are calculated from a nonlinear transformation as:

$$\alpha_{Ti} = \sigma(W_{Ti} f^{'}_{char} + U_{Ti} f^{'}_{word} + b_{Ti}) \tag{5}$$

$$\alpha_{Tc} = \sigma(W_{Tc} f^{'}_{char} + U_{Tc} f^{'}_{word} + b_{Tc}) \tag{6}$$

where σ is the sigmoid function, $W \in R^{d^{'} \times d^{'}}$ and $U \in R^{d^{'} \times d^{'}}$ are weight matrixes, and b is the bias term.

Then base these gates, two different vector would be separately calculated by:

$$f_{Ti} = \alpha_{Ti} \odot f^{'}_{char} + (1 - \alpha_{Ti}) \odot f^{'}_{word} \tag{7}$$

$$f_{Tc} = \alpha_{Tc} \odot f^{'}_{char} + (1 - \alpha_{Tc}) \odot f^{'}_{word} \tag{8}$$

where f_{Ti} and f_{Tc} are the hybrid features respectively for trigger identification and trigger classification; $\odot$ is the elementwise product operation.

3.4 Training and Classification

Following the above works, we consider event detection as a multi-class classification problem and use stochastic gradient descent for training. To train the trigger generator, we dispose the training datasets by using negative sampling. Concretely, we divide all characters in sentences into positive and negative samples according to whether the characters are included in triggers. Given the current token w_i, we feed its feature F_i into a fully-connected network to predict the trigger label as:

$$T_i = f(W_T F_i + b_T) \tag{9}$$

$$y_{t_i} = softmax(W T_i + b) \tag{10}$$

where f is a non-linear activation and y_{t_i} is the output of the $i-th$ trigger label.

Then we can define the objective function to train the trigger classifier as:

$$J(\theta) = \sum_{i=1}^{n} log(p(y_{t_i}|\theta)) \tag{11}$$

where n is the number of tokens and θ is the network parameter. We employ stochastic gradient descent to maximize the $J(\theta)$.

4 Experiments

4.1 Dataset, Evaluation Metric and Hyperparameter Setting

We evaluate our model on ACE 2005 dataset and KBPEval2017 dataset for event detection. To compare with previous work ([3,7] and [27]), we adopt the same data split to the previous work. According to this data split, there are 569 documents in train set, 64 documents in development set and 64 documents in test set. KBPEval2017 is another common dataset that is widely used for event extraction. In KBPEval2017 dataset, there are 693 documents that derived from newswire and discussion forum.

The experiments are designed to evaluate the performance of our model in trigger identification and trigger type classification. We calculate recall (R), precision (P) and F-measure (F) that same as Feng et al., 2016 to evaluate our model.

In the process of preprocessing, we use the Stanford CoreNLP toolkit to conduct tokening and generate dependency parsing trees. For negative sampling, the ratio of positive to negative instances is set to 1:20. In embedding layer, we respectively limit to 220 and 130 tokens in character-level representation and word-level representation. The dimensions of word embedding and character embedding are both 100, while the dimension of position embedding is set to 10. We use a three-layer GCN and a one-layer CNN with 400 filters. We also set batch size to 32 and dropout rate to 0.5.

4.2 Comparison with Baselines

For evaluating our method, we compare our method with the following baselines:

C-BiLSTM: A deep learning model for Chinese event extraction based on bidirectional LSTM and CNN [27]. C-BiLSTM views the event detection as a sequence labeling problem instead of relying on complicated natural language processing tools.

FBRNN: A forward-backward recurrent neural networks for event detection [8]. FBRNN is one of the first attempts to solve the problem of multi-word events in event detection.

HNN: A language-independent neural networks proposed by [7] to capture both sequence and structure information. HNN is a hybrid neural network which incorporates Bi-LSTM and convolutional neural networks for event detection.

NPN: A recent state-of-the-art deep learning model for Chinese event detection which respectively use three kinds of hybrid representation learning methods (concat, general and task-specific) to generate representation for event detection [15].

In these models, C-BiLSTM is a character-based model, while FBRNN and HNN are word-based model. Our model and NPN are based on both word and character. Table 1 presents the performance of the models on the test sets. As we can see from the table, character-based models like C-BiLSTM or word-based models like FBRNN and HNN have a relatively low F1 scores. It means that whether single word-based or single character-based models are not provide sufficient feature information for event detection. NPN performs better than HSGCN in recall value and HNN have a higher precision than NPN and HSGCN. However, HSGCN get competitive results on precision and recall that result in a highest F1 score among all models. Specifically, our model improve the F1 scores by 1.2% and 0.9% in trigger identification and trigger classification. What's more, the experiment results of HNN, NPN and HSGCN show that our model do not need sacrifice the recall to improve precision

Table 1. Experiment results on ACE 2005 dataset. (*) indicates the result taken from the original paper.

Model	Trigger identification (%)			Trigger classification (%)		
	P	R	F1	P	R	F1
C-BiLSTM	65.6	66.7	66.1	60.0	60.9	60.4
FBRNN	64.1	63.7	63.9	59.9	59.6	59.7
HNN	**74.2**	63.1	68.2	**77.1**	53.1	63.0
NPN (Task-specific)*	64.8	**73.8**	69.0	60.9	**69.3**	64.8
HSGCN	71.5	68.9	**70.2**	66.9	64.6	**65.7**

4.3 Performance on Specific Event Subtype

To further analyze the performance and effectiveness of HSGCN, we evaluate our model in different specific event subtypes. In this subsection, we evaluate HSGCN on KBPEval2017 dataset and test the model on five most common event subtype (Attack, Broadcast, Transport Ownership, Die and Transfer Money) of KBPEval2017 dataset. For comparison purposes, we select NPN (Task-specific) as baseline in this subsection. Table 2 show the specific result of HSGCN and NPN (Task-specific) in trigger identification and trigger classification.

As we can observe from the result, our model achieve better performance for almost every event subtype, either in trigger identification or trigger classification. The experimental results show the effectiveness of our model. Regarding the results of baseline on KBPEval2017 dataset, HSGCN is further improved, achieve 0.85% and 1.14% improvements of F1 score in trigger identification and trigger classification. In the same setting for datasets and performance measures, HSGCN gets better precision and F1 score, in spite of the model has a relatively lower recall. The same trends are observed when test HSGCN in specific event subtypes. As the result of Tables, we can deem both semantic and syntactic information are necessary in event detection. We believe that HSGCN has superior performance to capture semantic and syntactic information. Due to this improvement, HSGCN achieve a better performance in the experiments.

Table 2. The result of HSGCN and baseline in trigger identification and trigger classification. (ΔF1) is equal to the F1 score of HSGCN minus the F1 score of baseline.

Event type	Baseline (%)			HSGCN (%)			ΔF1
	P	R	F1	P	R	F1	
Trigger identification							
Attack	61.93	55.81	58.71	68.12	52.44	59.26	+0.55
Broadcast	65.29	48.20	55.46	68.21	48.81	56.90	+1.44
Transport ownership	66.79	45.22	53.93	70.10	46.25	55.73	+1.80
Die	64.17	55.45	59.49	66.93	54.49	60.07	+0.58
Transfer money	69.55	50.20	58.31	71.63	51.12	59.66	+1.35
All event type	63.72	53.56	58.20	67.91	52.24	59.05	+0.85
Trigger classification							
Attack	58.10	47.29	52.14	62.06	46.91	53.43	+1.29
Broadcast	60.07	42.55	49.81	63.57	43.65	51.76	+1.95
Transport ownership	55.75	48.76	52.02	59.85	47.78	53.14	+1.12
Die	62.83	49.73	55.52	67.68	47.53	55.84	+0.32
Transfer money	59.90	49.12	53.98	62.74	46.94	53.70	−0.28
All event type	56.97	47.68	51.91	63.65	45.47	53.05	+1.14

5 Conclusions

In this paper, we propose a hybrid syntactic graph convolutional networks (HSGCN) for Chinese event detection that exploit the syntactic information to effectively model the sentence-level feature. Our model also consider the different effects of words and characters in trigger identification and trigger classification. Compared with traditional event detection methods, our approach efficiently capture syntactic dependency and generate a sentence-level feature based on GCN, thus can take advantage of event information that scattered in the sentence. Furthermore, in order to keep the original semantic of Chinese vocabulary, word embedding and character embedding are concatenated, which improve the accuracy of triggers identification. Experimental results show that our method achieve a competitive performance on ACE 2005 and KBPEval2017 benchmark datasets.

Acknowledgements. This research was funded by the National Natural Science Foundation of China, grant number 61402220, the Philosophy and Social Science Foundation of Hunan Province, grant number 16YBA323, the Scientic Research Fund of Hunan Provincial Education Department for excellent talents, grant number 18B279, the key program of Scientic Research Fund of Hunan Provincial Education Department, grant number 19A439, the Project supported by the Natural Science Foundation of Hunan Province, China, grant number 2020JJ4525.

References

1. Ahn, D.: The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 1–8 (2006)
2. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750 (2014)
3. Chen, Z., Ji, H.: Language specific issue and feature exploration in Chinese event extraction. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 209–212 (2009)
4. Chieu, H.L., Ng, H.T.: A maximum entropy approach to information extraction from semi-structured and free text. In: AAAI/IAAI 2002, pp. 786–791 (2002)
5. Chung, T., Liu, Y., Xu, B.: Monotonic alignments for summarization. Knowledge-Based Syst. **192**, 105363.1–105363.10 (2020)
6. Chung, T., Xu, B., Liu, Y., Ouyang, C., Li, S., Luo, L.: Empirical study on character level neural network classifier for Chinese text. Eng. Appl. Artif. Intell. **80**, 1–7 (2019)
7. Feng, X., Qin, B., Liu, T.: A language-independent neural network for event detection. Sci. China Inf. Sci. **61**(9), 092106 (2018)
8. Ghaeini, R., Fern, X.Z., Huang, L., Tadepalli, P.: Event nugget detection with forward-backward recurrent neural networks. arXiv preprint arXiv:1802.05672 (2018)
9. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1127–1136. Association for Computational Linguistics (2011)
10. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. In: Proceedings of ACL-2008: HLT, pp. 254–262 (2008)
11. Jin, H., Li, C., Zhang, J., Hou, L., Li, J., Zhang, P.: XLORE2: large-scale cross-lingual knowledge graph construction and application. Data Intell. **1**(1), 77–98 (2019)
12. Jin, Y., Kan, M.Y., Ng, J.P., He, X.: Mining scientific terms and their definitions: a study of the ACL anthology. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 780–790 (2013)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Li, P., Zhou, G., Zhu, Q., Hou, L.: Employing compositional semantics and discourse consistency in Chinese event extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1006–1016 (2012)
15. Lin, H., Lu, Y., Han, X., Sun, L.: Nugget proposal networks for Chinese event detection. arXiv preprint arXiv:1805.00249 (2018)
16. Liu, Y., Ouyang, C., Li, J.: Ensemble method to joint inference for knowledge extraction. Expert Syst. Appl. **83**, 114–121 (2017)
17. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. arXiv preprint arXiv:1703.04826 (2017)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

19. Nguyen, T.H., Grishman, R.: Modeling skip-grams for event detection with convolutional neural networks. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 886–891 (2016)
20. Nguyen, T.H., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: Thirty-second AAAI conference on artificial intelligence (2018)
21. Rudinger, R., White, A.S., Van Durme, B.: Neural models of factuality. arXiv preprint arXiv:1804.02472 (2018)
22. Sha, L., Qian, F., Chang, B., Sui, Z.: Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
23. Vashishth, S., Bhandari, M., Yadav, P., Rai, P., Bhattacharyya, C., Talukdar, P.: Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. arXiv preprint arXiv:1809.04283 (2018)
24. Wan, H., Zhang, Y., Zhang, J., Tang, J.: AMiner: search and mining of academic social networks. Data Intell. **1**(1), 58–76 (2019)
25. Westerhout, E.: Definition extraction using linguistic and structural features. In: Proceedings of the 1st Workshop on Definition Extraction, pp. 61–67. Association for Computational Linguistics (2009)
26. Yubo, C., Liheng, X., Kang, L., Daojian, Z., Jun, Z., et al.: Event extraction via dynamic multi-pooling convolutional neural networks (2015)
27. Zeng, Y., Yang, H., Feng, Y., Wang, Z., Zhao, D.: A convolution BiLSTM neural network model for Chinese event extraction. In: Lin, C.-Y., Xue, N., Zhao, D., Huang, X., Feng, Y. (eds.) ICCPOL/NLPCC-2016. LNCS (LNAI), vol. 10102, pp. 275–287. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50496-4_23
28. Zhang, J., Qin, Y., Zhang, Y., Liu, M., Ji, D.: Extracting entities and events as a single task using a transition-based neural model. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 5422–5428. AAAI Press (2019)
29. Zhang, T., Ji, H., Sil, A.: Joint entity and event extraction with generative adversarial imitation learning. Data Intell. **1**(2), 99–120 (2019)