



# Event Detection Based on Multilingual Information Enhanced Syntactic Dependency GCN

Zechen Wang<sup>1</sup>, Binbin Li<sup>2</sup>(✉), and Yong Wang<sup>2</sup>(✉)

<sup>1</sup> School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

wangzechen@iie.ac.cn

<sup>2</sup> Institute of Information Engineering, CAS, Beijing, China  
{libinbin,wangyong}@iie.ac.cn

**Abstract.** Event detection is a hot and difficult problem in information extraction. It is widely used in automatic news extraction, financial event analysis and other fields. However, most of the existing event detection methods only focus on a single language, ignoring the event information provided by other languages, and can not solve the problem of polysemy in a single language, which makes it difficult to improve the performance of event detection methods. To solve these problems, this paper proposes a new Event Detection based on Multilingual Information Enhanced Syntactic Dependency GCN. Specifically, the model translates the original language and aligns words, takes multilingual data as input, and constructs syntactic dependency diagrams for initial language sentences. Then, a graph neural network is constructed based on the syntactic dependency graph, and combined with the attention mechanism, the nodes of the syntactic dependency graph are enhanced by the translated language. Finally, the classifier finds the trigger and judges the event type. The model effectively improves the recognition efficiency of polysemous words by using multilingual information, and makes full use of sentence structure information by using syntactic dependency graph. Experiments on ace2005 benchmark data set show that the model can detect events effectively and is obviously superior to the existing event detection methods.

**Keywords:** Event detection · Multilingual information · Semantic dependency graph · GCN · Attention mechanism

## 1 Introduction

Event detection belongs to information extraction task, which is an important natural language processing task. The purpose is to identify the event reference from the text and determine the category of the event [2]. Specifically, for a given sentence, it is necessary to detect whether there are triggers in the sentence and classify the triggers. [7] at present, there are still many problems in event detection. This paper mainly solves the following two problems:

First, in the event extraction corpus similar to ace2005 data set [5], the single language corpus often lacks effective information to distinguish the ambiguity of polysemy. For example, for the sentence “an American tank fired on the abandoned hotel”. It is necessary to detect and extract “fire” as the trigger of the event contained in the marking sentence, and classify the event according to the content described by the event. Obviously, since the trigger “fire” means “shot”, this sentence expresses an attack event. According to the classification of events in ACE2005 guidance document, the event should be classified as “attack”. However, in the process of automatic extraction, the word “fire” may be incorrectly recognized. For example, in sentence 2, “he has fired his air defense chief”, “fire” means “dismissal”, so it will be wrongly classified as “end position” according to ace2005 guidance document. Fortunately, a polysemy in a given language often corresponds to multiple monosemy in another language. With the development of machine translation tasks in recent years, the method of machine translation can translate polysemy more accurately by combining context and other information.

Second, the existing event detection methods often do not fully analyze the syntactic structure. A sentence is a sequence of words. It is generally believed that the closer the distance between words, the greater the relevance. Verbs, nouns and adjectives are more likely to appear in sentences as triggers. However, compared with the distance and part of speech between words, the direct or indirect relationship of words in sentence structure is more important to identify triggers. For the sentence “an American tank fired on the abandoned hotel”. In the process of automatic extraction, the word “abandoned”, as a verb, may also be mistakenly regarded as a trigger, resulting in the error of trigger recognition and event type judgment. In order to correctly distinguish the relationship between different verbs and nouns in sentences, dependency parsing is often used. In recent years, the types of dependency parsing methods have gradually increased. Each has its own advantages and disadvantages. It can label sentences with simple or complex structures, and gradually expand its application in a variety of natural language processing tasks.

Based on the existing research, this paper proposes an event detection method based on multilingual information enhanced syntactic dependency GCN, which can make good use of syntactic structure and multilingual information. The model translates the original language, constructs a graph convolution network based on syntactic dependency graph, solves the ambiguity of monolingual words, fully extracts the relationship between words, and finally finds the trigger accurately through the classifier to judge the event type. Finally, by comparing with baseline experiment, the superiority of this method in accuracy and F1 value can be reflected.

## 2 Related Works

Previously, there have been some research on event detection based on multilingual enhancement and dependency parsing.

For multilingual enhancement, Zhu et al. [21] Proposed a Chinese English event extraction model. However, the model uses traditional machine learning

methods to extract features, and can not deeply analyze the structure of structured sentences. Liu et al. [14]. Proposed a cross language event detection method. This method shows high operation efficiency when dealing with articles containing multiple languages, but the latest and most effective translation tools are not fully utilized, and only achieve general results when allowing longer operation time. Chen et al. [15] Proposed to realize the event detection task based on multilingual gated attention mechanism and LSTM. This method also uses multilingual information to solve the problem of polysemy at one time. However, LSTM focuses more on the sequence information of context and lacks the semantic association information between words.

Some natural language processing models have used various features such as vocabulary, grammar and semantics as input for event detection. For example, Liu et al. [16] believed that triggers and arguments should be paid more attention than other words in the process of event detection, so they constructed an attention vector to encode each trigger, argument and context word. EDEEI model [20] constructs a part of speech based attention map, which uses the correlation between part of speech and trigger text to capture events. These methods only use part of speech and location to construct the network, and do not really use dependency syntax to analyze the relationship between words. Dependency parsing based methods are widely used in the field of biology. Kilicoglu proposed heuristic [8] and trigger based [9] methods. These methods need to construct grammatical rules for biological events, which are difficult to be widely used in news and other texts. Lai et al. [11] Constructed a graph neural network for biological texts based on dependency parsing. The generality of the model is greater than that of previous studies. However, in order to increase the computational efficiency, the node information is simplified by scoring, and the node information is over compressed.

To sum up, there are still many problems to be solved in the research of event extraction based on multilingual enhancement and dependency parsing.

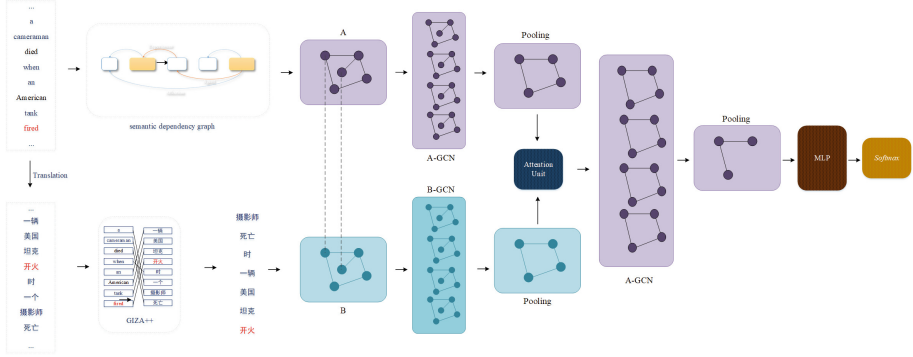
### 3 Contribution

The following contributions differentiate our method from previous work.

1. A graph neural network structure based on dependency syntactic graph is designed. By constructing syntactic graph, we can better capture the dependencies between words, and capture the relationship between these relationships and triggers through GCN.
2. Based on the constructed graph neural network, a multi language node enhancement method based on word alignment and attention mechanism is proposed to solve the problem of word ambiguity through multi language comparison
3. The evaluation of the proposed method on the ace2005 benchmark data set shows that the proposed method has better performance than other latest methods.

## 4 Method

In this section, we present our framework for the proposed Event Detection based on Multilingual Information Enhanced Syntactic Dependency GCN (MS-GCN) model. We first describe the hierarchy of the model, and then show the details of the algorithm along with the key intuition underlying it.



**Fig. 1.** The framework of MS-GCN model.

The proposed framework is illustrated in Fig. 1. The event detection can be treated as a classification problem in the proposed model which detects events and event types by identifying triggers and trigger types. This section will introduce the framework of the model, first describe the hierarchy of the model, and then show the details of the algorithm. The framework of MS-GCN model is shown in Fig. 1. Similar to the existing research methods, MS-GCN model also solves the problem of event detection as a word classification problem. The model traverses each word in the sentence to determine whether it is a trigger. If so, it further determines which event type the word represents. MS-GCN model includes the following parts: Translation, Multilingual word alignment, dependency syntax graph generation, GCN construction, pooling, node attention calculation, secondary pooling, classification.

Text translation obtains the multilingual text corpus corresponding to the original event detection corpus through the method of machine translation, and uses the word alignment tool to establish a one-to-one mapping relationship for the words in the translated corpus. Connecting these vectors can generate a new word vector. The newly generated word vector is used for feature extraction, node enhancement and feature selection. Node enhancement extracts original features from feature extraction, and provides processed features for feature selection to obtain high-quality features. Finally, the feature is input into the classifier to get the trigger and its classification. Each part of the model is described in detail below.

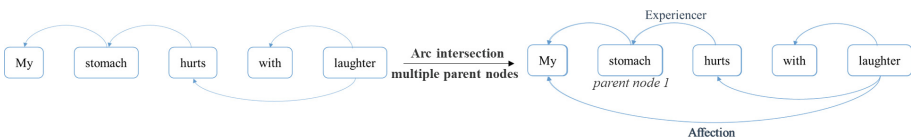
### 4.1 Multilingual Alignment

MS-GCN model calls the existing Baidu machine translation service for text translation. Take ace2005 English text as input and output the corresponding Chinese translation text. The translated Chinese text is segmented, and Giza ++ [17] is used to align the text before and after translation. Giza ++ is a widely used word alignment program, which is generally applied to phrase based translation systems. In the process of word alignment using Giza ++, firstly, unsupervised hidden Markov models (HMM) are trained based on Baum Welch method, and these models are used to generate Viterbi alignment between bilingual words or phrases [19].

During word alignment training, in order to solve the problem of small sample size of event detection data set and improve the accuracy of word alignment, MultiUN [3] data set is spliced with event detection data set and translation corpus of corresponding language to increase the total amount of training data. MultiUN dataset is suitable as an extended corpus because its translation results have been manually verified, including 7 languages, 21 bitexts, 489334 files and 1.99Gb Tokens. According to the word alignment results, the word order of the translated corpus text is adjusted in the sentence, so that the word order of the translated text is the same as that of the original text as much as possible. As shown in the example in Fig. 1, the original English text is “cameraman died when an American tank fired”, the translated text is “一辆美国坦克开火时摄影师死亡”, and the text after word segmentation and word alignment is “摄影师死亡时一辆美国坦克开火”.

### 4.2 Dependency Parsing Feature

See Fig. 2.



**Fig. 2.** Comparison of dependency tree (left) and dependency graph (right).

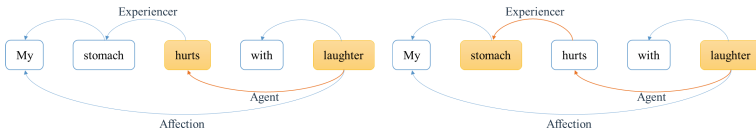
Dependency parsing (DP) reveals its syntactic structure by analyzing the dependency between components in a language unit. [] intuitively speaking, dependency parsing identifies the grammatical components of “subject predicate object” and “definite complement” in the sentence, and analyzes the relationship between each component. At present, dependency semantic tree is widely used for dependency syntactic analysis. However, the form of dependency tree often omits some important semantic relationships. Semantic dependency graph

parsing allows arc intersection and multiple parent nodes on the basis of semantic dependency tree, which makes the analysis of grammatical structures such as conjunction, concurrent language and conceptual transposition more comprehensive (Table 1).

**Table 1.** 16 dependency semantic relations.

Tag	SBV	VOB	IOB	FOB
Description	Subject-verb	Verb-object	Indirect-object	Fronting-object
Tag	COO	POB	LAD	RAD
Description	Coordinate	Preposition-object	Left adjunct	Right adjunct
Tag	DBL	ATT	ADV	CMP
Description	Double	Attribute	Adverbial	Complement
Tag	IS	WP	HED	NONE
Description	Independent structure	Punctuation	Head	None

We select 16 dependency semantic relations for annotation, including 14 kinds of relevance and header (HED) and non relevance (none) (Fig. 3).



**Fig. 3.** Direct relationship (left) and indirect relationship (right).

The main structure of a general sentence contains one or two subjects and is associated with a trigger. Therefore, direct correlation and indirect correlation are selected for corpus statistics. There are 15 \* 15 possible relationships between the two words. Generate a dependency syntax matrix with a size of 225 \* n (n is the maximum sentence length), and statistically generate an association representation matrix by counting the relationship between the semantic dependency graphs corresponding to each sentence. Then the matrix is compressed by SVD and normalized to obtain the vector representation of each relationship. The resulting semantic dependency feature vector can be expressed as a combination of dependency vector and numerical representation of the relative position of relational words, which is represented as SDF (Fig. 4).

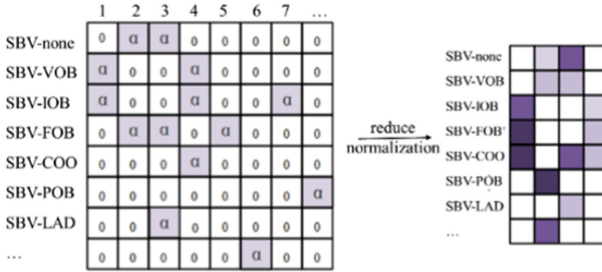


Fig. 4. Generation of SDF.

### 4.3 Node Vector Representations

In this paper, node vector of GCN is composed of three feature vectors: content word feature vector (CWF), position feature vector (PF) and dependent syntactic feature vector (DPF). Among them, CWF is a word vector, and each word corresponds to a CWF vector, which can distinguish the meaning of the same word in different contexts. Pf reflects the position of triggers, counting from the first word of each sentence. The position information is expressed as an integer and further transformed into a unique heat vector. DPF is the dependency syntactic feature vector introduced in the previous section.

The word vector used for word representation in this paper is generated after fine tuning Bert [8] Based on the training corpus. A new vector structure is further constructed based on word vector, which is spliced by CWF and PF. MS-GCN model is improved and fine tuned based on the model. Ace2005 is used as the fine tuning training data set to train the fine tuned word vector by completing the sentence classification task. Through this fine-tuning training, the produced word vectors generate different word vectors for the same word corresponding to different contexts, which can distinguish the different meanings of words with the same spelling in different contexts, so as to solve the problem of polysemy. At the same time, through the pre-training of large corpus, a large amount of external information is introduced to supplement the information not contained in the context of event detection task corpus, which effectively solves the problem of insufficient information caused by the small event detection corpus.

Position vectors are used to represent the position information of words in sentences. In the process of event detection, it is necessary to classify the words in the input sentence. In order to express the trigger information in a sentence, it is necessary to establish the relationship between each word in the sentence and the candidate trigger. To construct this relationship, PF is defined as the relative distance between the current word and the candidate trigger. PF is encoded, and each distance value is represented by an embedded vector. When training the distance vector, we need to construct the matrix to generate the distance vector, initialize and optimize it.

Let the size of CWF be  $d_{\text{CWF}}$ , the size of SF be  $d_{\text{SF}}$ , the size of SDF be  $d_{\text{SDF}}$ , the size of location code be  $d_{\text{PF}}$ . Represent word vector of the  $i$ -th word in the sentence as  $x_i \in R^d$ ,  $d = d_{\text{CWF}} + d_{\text{SF}} + d_{\text{SDF}} + d_{\text{PF}} * 2$ .

#### 4.4 GCN Construction

We construct this graph convolutional network models as an undirected connected graph [10]  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ . Which consists of a set of nodes  $|\mathcal{V}|$  with  $|\mathcal{V}| = n$ , a set of edges  $|\mathcal{E}|$  with  $|\mathcal{E}| = n$  and the adjacency matrix  $|\mathcal{A}|$ . If there is an edge between node  $|i\rangle$  and node  $|j\rangle$ , the entry  $\mathbf{A}(i, j)$  denotes the weight of the edge; otherwise,  $\mathbf{A}(i, j) = 0$ . We denote the degree matrix of  $\mathbf{A}$  as a diagonal matrix  $\mathbf{D}$ , where  $\mathbf{D}(i, i) = \sum_{j=1}^n \mathbf{A}(i, j)$ . Then, the Laplacian matrix of  $\mathbf{A}$  is denoted as  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ . The corresponding symmetrically normalized Laplacian matrix is  $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ , where  $\mathbf{I}$  is an identity matrix.

The adjacency matrix corresponding to the source language is represented as  $\mathbf{A}$ , and the adjacency matrix represented by the translated language is  $\mathbf{B}$ . When calculating the graph convolution for the first time, steps  $\mathbf{A}$  and  $\mathbf{B}$  are the same. For the second time, we just calculate on  $\mathbf{A}$ . Here, take  $\mathbf{A}$  as an example. This deep model on graphs contains several spectral convolutional layers that take a vector  $\mathbf{X}^p$  of size  $n \times d_p$  as the input map of the  $p$ th layer and output a map  $\mathbf{X}^{p+1}$  of size  $n \times d_{p+1}$  by:

$$\mathbf{X}^{p+1}(:, j) = \sigma \left( \sum_{i=1}^{d_p} \mathbf{V} \begin{bmatrix} (\theta_{i,j}^p)(1) & 0 \\ & \ddots \\ 0 & (\theta_{i,j}^p)(n) \end{bmatrix} \mathbf{V}^T \mathbf{X}^p(:, i) \right), \quad \forall j = 1, \dots, d_{p+1}$$

where  $\mathbf{X}^p(:, i)$  ( $\mathbf{X}^{p+1}(:, j)$ ) is the  $i$ th ( $j$ th) dimension of the input (output) map, respectively;  $\theta_{i,j}^p$  denotes a vector of learnable parameters of the filter at the  $p$ th layers. Each column of  $V$  is the eigenvector of  $L$  and  $\sigma(\cdot)$  is the activation function.

#### 4.5 Node Enhancement

The node enhancement contains an attention unit mainly contains the attention node enhancement module. Attention mechanism is usually used to reweight and encode vector sequences. In the MS-GCN model, the bilingual logical unit uses the attention mechanism to emphasize the relationship between different words expressing the same meaning in the two languages. The node enhancement module pairs the maps corresponding to Chinese and English sentences as the input of attention mechanism. The word meaning of each candidate trigger is directly represented by word vectors from two different languages, so as to emphasize the word meaning of the trigger to be extracted and realize the disambiguation of polysemy.

Each map generated in feature extraction module is a  $n \times X$ . The maps represented as  $K$  are taken as the inputs of attention mechanism. The attention calculation process is as follows. A new random matrix  $WQ$  of length  $w$  is computed. The product of two vectors is calculated to obtain a new matrix  $Q$ .



The random matrix  $WK$  whose width is  $w$  and the length is  $k_1$  is acquired, and the product of the random matrix  $WK$  and  $WQ$  produces the matrix  $WV$ . Calculate the product of  $WV$  and map to gain the matrix  $V$ .

Based on the three generated  $K, Q, V$  matrices, an attention matrix  $Z$  is calculated by using the following formula:

$$Z = \text{softmax} \left( \frac{Q \times K^T}{\sqrt{X}} \right) V$$

Train  $Wk, WQ, WV$  matrices. The scoring function is as follows:

$$f_{\text{score}} = \frac{Q \cdot K^T}{\sqrt{X}}$$

Then the  $Z$  matrix is compressed with max pooling to generate a vector  $z$ . Based on the updated  $WK, WQ$  and  $WV$ , the product of  $z$  and  $K$  constructs a new attention map.

## 4.6 Classifier

This module concatenates the CWFs of the current word and the words on the left and right of the current one, to obtain the vector  $P$  of length  $3 * CWF$ . The learned sentence level features and word features are connected into a vector  $F = [L, P]$ . In order to calculate the confidence of the event type of each trigger, the feature vector is inputted into the classifier  $O = W_s F + b_s$ .  $W_s$  is the transformation matrix of the classifier,  $b_s$  is the bias, and  $O$  is the final output of the network, where the output type is equal to the total number of event types plus one to include the “not a trigger” tag that does not play any role in the event.

## 5 Experiment

In this section, we design three different scenarios based on ACE 2005 benchmark dataset for event detection. We investigate the empirical performances of our model and compare it to the existing state-of-the-art models. The ACE 2005 dataset is utilized as the benchmark experimental dataset. The test set used in the experiment contains 40 Newswire articles and 30 other documents randomly selected from different genres. The remaining 529 documents are used as the training set.

### 5.1 Experimental Settings

On Wikipedia and bookcorpus, BERT is trained to generate the word content vector. The dimension of the CWF is set as 128. WordNet 3.0 is utilized to generate SF, the number of words used in training is 6 thousand and the dimension of word vector structure is 488.

In trigger classification, the window size is 3. We set the number of convolution kernel to 200, batch size to 170, and position vector dimension to 5.

Random gradient descent is used to train the neural network. It mainly includes two parameters  $p$  and  $\alpha$ . Set  $p = 0.95$  and  $\alpha = 1\text{E-}6$ . For drop out operations, set the rate to 0.5. The optimizer is Adam.

Similar to the previous work, we use the following criteria to judge the correctness of each predicted event. The trigger recognition is correct if the extracted trigger matches the reference trigger. The recognition and classification of the trigger are correct if the event subtype of the extracted trigger matches the event subtype of the reference trigger.

Based on the above criteria, the effect of event detection is judged, and Precision (P), Recall (R) and F value (F1) are used as evaluation indexes.

## 5.2 Evaluation of Event Detection Methods

To demonstrate how the proposed algorithm improves the performance over the state-of-the-art event detection methods, we compare the following representative methods from the literature:

- (1) Li's baseline [12]: Li et al. proposed a feature-based system which used artificially designed lexical features, basic features and syntactic features.
- (2) Liao's cross-event [13]: The cross-event detection method proposed by Liao and Grishman used document level information to improve the performance of ACE event detection.
- (3) Hong's cross-entity [6]: Hong et al. exploited a method to extract events through cross-entity reasoning.
- (4) Li's joint model [12]: Li et al. also developed an event extraction method based on event structure prediction.
- (5) DMCNN method [1]: A word representation model was established to capture the semantic rules of words, and adopted a framework based on dynamic multi pool convolutional neural network.
- (6) EDEEI method [20]: an event detection method based on external information and semantic network adopts the neural network framework including part of speech and attention map (Table 2).

**Table 2.** Overall performance on the ACE 2005 blind test data

Methods	Trigger identification			Trigger classification		
	P	R	F	P	R	F
Li's baseline	76.2	60.5	67.4	74.5	59.1	65.9
Liao's cross-event	N/A			68.7	68.9	68.8
Hong's cross-entity	N/A			72.9	64.3	68.3
Li's joint model	76.9	65	70.4	73.7	62.3	67.5
DMCNN model	80.4	67.7	73.5	75.6	63.6	69.1
EDEEI model	<b>77.0</b>	<b>72.9</b>	<b>74.9</b>	<b>77.3</b>	<b>62.2</b>	<b>69.9</b>
MS-GCN model	<b>78.5</b>	<b>71.8</b>	<b>75.0</b>	<b>78.2</b>	<b>63.7</b>	<b>70.2</b>

Among all methods, MS-GCN model has the best performance. Compared with the existing methods, the accuracy and F value of trigger recognition are significantly improved. Compared with Li, Liao and Hong’s methods, it can be found that only relying on vocabulary, syntax and features is not enough to accurately extract triggers. The comparison with DMCNN shows that the semantic rules that can be captured only by the word representation model are relatively limited. The comparison with EDEEI model shows that the attention mechanism constructed only by part of speech information is lower than MS-GCN model in distinguishing ambiguous words. The introduction of multilingual knowledge can effectively improve the accuracy of event detection.

### 5.3 Analysis of Different Languages

This section presents a detailed comparison of the translation attention between en-de, en-fr and en-cn respectively. The purpose is to test for advantages and disadvantages of each language pair.

The advantages of using en+cn can be observed visually and quantitatively in Table 3. It can be seen that the combination of English and Chinese achieves the best performance on both trigger identification and trigger classification. It may be because that Chinese has more different syntax than french and Germany.

**Table 3.** Performance with different languages.

Methods	Trigger identification F (%)	Trigger identification + classification F (%)
en+de	66.2	56.9
en+fr	71.7	66.7
en+cn	<b>75.0</b>	<b>70.2</b>

### 5.4 Effectiveness of Semantic Dependency Graph Features

In order to verify the effectiveness of attention mechanism, similar to the method used in literature [4, 18], this paper conducted a comparative experiment with and without dependent syntactic features. It can be seen from Table 4 that the model with dependent syntactic features is better than the model without dependent syntactic features in event detection.

**Table 4.** Performance with and without semantic dependency graph features

Methods	Trigger identification F (%)	Trigger identification + classification F (%)
Without SDF	73.2	65.9
With SDF	<b>75.0</b>	<b>70.2</b>

Experimental results show that dependency syntactic features improve the efficiency of event detection. It shows that the syntactic map successfully establishes the deep relationship between words, and the characteristics of this relationship are successfully extracted. This relationship is helpful to improve the effect of trigger recognition and classification.

## 6 Conclusion

This paper proposes an event detection method based on multilingual information enhancement and syntactic dependency graph. This paper designs a GCN model based on syntactic dependency graph, constructs an attention mechanism based on multilingual information, and makes the syntactic features related to triggers easier to capture. Experiments on the widely used ace2005 benchmark data set show that this method is obviously superior to the existing event detection methods. In addition, the experimental results are fully analyzed in this paper. By showing the performance of the algorithm, it is proved that MS-GCN is a very effective event detection model

**Acknowledgments.** This work was supported by National Natural Science Foundation of China (No. 61931019).

## References

1. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: ACL, pp. 167–176 (2015)
2. Cheng, D., Yang, F., Wang, X., Zhang, Y., Zhang, L.: Knowledge graph-based event embedding framework for financial quantitative investments. In: SIGIR 2020: The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
3. Eisele, A., Yu, C.: Multium: a multilingual corpus from united nation documents. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta (2010)
4. Ferguson, J., Lockard, C.: Semi-supervised event extraction with paraphrase clusters. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (2018)
5. Grishman, R., Westbrook, D., Meyers, A.: NYU’s English ACE 2005 system description. J. Satisfiability (2005)
6. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: ACL, pp. 1127–1136 (2011)
7. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. In: ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 15–20 June 2008, Columbus, Ohio, USA (2008)
8. Kilicoglu, H., Bergler, S.: Syntactic dependency based heuristics for biological event extraction. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 119–127. Association for Computational Linguistics, Boulder, Colorado, June 2009

9. Kilicoglu, H., Bergler, S.: Effective bio-event extraction using trigger words and syntactic dependencies. *Comput. Intell.* **27**(4), 583–609 (2011)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016)
11. Lai, V.D., Nguyen, T.N., Nguyen, T.H.: Event detection: gate diversity and syntactic importance scores for graph convolution neural networks (2020)
12. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: *ACL*, pp. 73–82 (2013)
13. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: *ACL*, pp. 789–797 (2010)
14. Liu, J., Chen, Y., Liu, K., Zhao, J.: Neural cross-lingual event detection with minimal parallel resources. In: *EMNLP-IJCNLP* (2019)
15. Liu, J., Chen, Y., Liu, K., Zhao, J.: Event detection via gated multilingual attention mechanism. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
16. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017)
17. Och, F.J.: Giza++: training of statistical translation models (2000)
18. Saedi, C., Branco, A., Rodrigues, J.A., Silva, J.: Wordnet embeddings. In: *Proceedings of the Third Workshop on Representation Learning for NLP, ACL 2018* (2018)
19. Tomeh, N., Allauzen, A., Yvon, F.: Maximum-entropy word alignment and posterior-based phrase extraction for machine translation. *Mach. Transl.* **28**(1), 19–56 (2013). <https://doi.org/10.1007/s10590-013-9146-4>
20. Wang, Z., Wang, S., Zhang, L., Wang, Y.: Exploiting extensive external information for event detection through semantic networks word representation and attention map. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) *ICCS 2021. LNCS*, vol. 12742, pp. 707–714. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-77961-0\\_56](https://doi.org/10.1007/978-3-030-77961-0_56)
21. Zhu, Z., Li, S., Zhou, G., Rui, X.: Bilingual event extraction: a case study on trigger type determination. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2014)