



# OPEN Joint event extraction model based on dynamic attention matching and graph attention networks

Jiajun Cheng<sup>1</sup>, Wenjie Liu<sup>2,3,4✉</sup>, Zhifan Wang<sup>2</sup>, Zhijie Ren<sup>2</sup> & Xingwen Li<sup>2</sup>

Event extraction is one of the important processes in event knowledge graph construction. However, extant event extraction models are confronted with the challenge of handling vague and unfamiliar event trigger words as well as noise that is prevalent in text. To address this issue, this study proposes a joint event extraction model that leverages dynamic attention matching and graph attention network. Specifically, the dynamic attention matching mechanism is employed to identify event nodes that contain text event structure features and to integrate event structure knowledge for constructing event pattern subgraph that correspond to the text, thereby resolving the problem of ambiguous and unknown trigger word classification. To better discriminate between semantic information and event structure information and to mitigate the impact of noise in text, we introduce a graph attention network that integrates event structure features for aggregating feature embedding of node neighbors. Experiment results on the ACE2005 dataset demonstrate that our proposed model attains competitive performance in comparison to existing methods.

Event extraction is one of the important components of the event knowledge graph construction process<sup>1,2</sup>. In the field of event extraction, the Automatic Content Extraction Evaluation Conference (ACE)<sup>3,4</sup> is undoubtedly the most influential evaluation event. The conference set a clear benchmark for event extraction, especially two key tasks of event extraction: event trigger word extraction and event argument extraction<sup>5,6</sup>. These two tasks are often treated as classification tasks in text processing. For event trigger word extraction, the main goal is to accurately identify the core words that can trigger specific events from a given text sequence, and further determine the event types represented by these words<sup>7,8</sup>. The realization of this process is crucial for accurately understanding the event content in the text. Event argument extraction is to further analyze and mark the role of each argument from the identified events. Arguments are the various components that make up an event. They each play different roles and together form a complete event description. Event argument extraction allows people to have a deeper understanding of the details and background of the event, thereby more comprehensively grasping the information in the text.

Despite the remarkable progress in event extraction achieved by recent neural network methods<sup>9,10</sup>, two significant challenges still exist, namely, ambiguous and unknown trigger words. Ambiguous trigger words refer to words that can trigger different event types depending on the context, owing to the diversity and ambiguity of the natural language. To address the challenges posed by trigger ambiguity, numerous studies are working on extracting contextual knowledge from event arguments, e.g., enhancing the model's understanding of the event by mining the deeper features of the arguments<sup>11–14</sup>, employing attention mechanisms to fully utilise contextual information<sup>15</sup>, and exploring the associations between the arguments' positional information<sup>16</sup>. Even if the event types of such fuzzy trigger words are correctly classified, existing event extraction models may still find it challenging to extract unknown trigger words that are not present in the training set. To address this problem, Lu et al.<sup>17</sup> proposed a delta-learning strategy to extract specific vocabulary and contextual knowledge to extract generalize knowledge for discriminating unseen triggers. Yi et al.<sup>18</sup> identified such unseen triggers by cosine scores of input sentences, trigger words and parameters in the training set.

Existing models primarily employ dynamic matching strategies to calculate the cosine scores of input sentences, trigger words, and arguments in the training set. They perform dynamic matching of event patterns and use Graph Convolutional Networks (GCN) to aggregate event structure and text semantic information

<sup>1</sup>School of Information Engineering, Huzhou University, Huzhou 313000, China. <sup>2</sup>School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China. <sup>3</sup>Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China. <sup>4</sup>Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services, Nanjing University of Information Science and Technology, Nanjing 210044, China. ✉email: wenjiel@163.com

for event extraction<sup>18</sup>. However, using only cosine scores as node weights cannot effectively distinguish the importance of different event patterns, and it fails to address the noise problem. Additionally, GCN aggregates the information of surrounding nodes equally and cannot effectively distinguish the importance of different nodes, which adversely affects the final event extraction performance.

To address the aforementioned issues, this paper proposes a joint event extraction model that utilizes a dynamic attention matching mechanism and Graph Attention Network (DAT-GAT) to aggregate event structure and text semantic information. This model aims to enhance the recognition and classification performance of trigger words and arguments. The proposed model first extracts the top-k event patterns that best match the text using the dynamic attention matching mechanism. Corresponding event pattern subgraphs are then constructed, and attention scores of different event patterns are calculated. Next, a graph attention network is employed in conjunction with event structure features to aggregate feature embeddings of node neighbors and event structure and textual semantic information. Finally, the self-attention mechanism-based trigger word classifier and argument classifier are utilized to predict the trigger word type and argument role. The proposed joint event extraction model leverages a dynamic attention matching mechanism and GAT to improve the recognition and classification performance of trigger words and arguments, and provides a potential solution to the problem of ambiguous and unknown trigger words, which can significantly enhance the accuracy and robustness of event extraction.

In summary, this paper makes several significant contributions, which are outlined as follows:

- We present a novel dynamic attention matching mechanism that enables the identification of candidate trigger words and arguments in the input sentence, thereby facilitating the construction of an event pattern subgraph that incorporates event structure knowledge into the semantic aggregation process.
- We introduce a graph attention network integrating event structure features that aggregates node features, which effectively distinguishes the relative importance of semantic and event structure information while simultaneously mitigating the impact of noise in text.
- By combining the aforementioned methods, we propose a joint event extraction model based on dynamic attention matching and graph attention network (GAT). Empirical evaluations on the ACE2005 dataset demonstrate that our model outperforms existing state-of-the-art models in event extraction. The remainder of this paper is organized as follows: Section [Related work](#) introduces the current research status of event extraction, Section [Methodology](#) provides the structure and specific details of the model, Section [Experiments](#) gives the experimental results and analysis, and Section [Conclusion](#) is the conclusion of the paper.

## Related work

This section provides an overview of the current state of research on event extraction.

### Event extraction

Since the emergence of pre-trained language models, joint event extraction methods have made significant progress. In 2018, Liu et al.<sup>19</sup> proposed the joint event extraction framework JMEE, which utilizes syntax trees to enhance information flow within sentences. This framework combines graph convolutional networks and self-attention mechanisms to effectively aggregate information about trigger words and argument elements, achieving new breakthroughs in joint event extraction. Wadden et al.<sup>20</sup> used bidirectional RNN to learn word embeddings and constructed a generation graph based on BERT's encoding of sentences, further promoting the research progress of event extraction. This method achieves accurate extraction of events by integrating adjacent contextual information and passing it to all tasks through FCN and softmax-based classifiers.

Subsequently, Lin et al.<sup>21</sup> proposed a new idea, utilizing BERT embeddings and FCN-based classifiers to calculate local scores for trigger words and argument roles in the input information. Through this score, the dependency relationship between trigger words and argument roles can be accurately reflected, thereby obtaining the best event extraction results. Du et al.<sup>22</sup> pioneered the event extraction task in the form of question and answer (QA), which realizes end-to-end event parameter extraction and can extract event parameters of characters that did not appear in training, broadening the application scope of event extraction. Su et al.<sup>23</sup> studied the problem of low recall rate in biomedical event extraction tasks and proposed an innovative end-to-end multi-task method. This method enhances the generalization ability of the model by sharing neural encoders. And the neural network is used to combine event parameters, which effectively improves the precision and recall rate of extraction. Li et al.<sup>24</sup> proposed a novel approach for event extraction, which utilizes reinforcement learning and incremental learning methods guided by dialogue. The proposed method addresses the challenge of identifying the changeable roles of event participants. By leveraging a task-oriented dialogue system, this approach gradually extracts multiple event parameters by utilizing the relationships between them. Additionally, the interaction between parameters is utilized to assist in determining the role of each parameter, thus improving the accuracy of event extraction.

In 2023, research on event extraction technology is expected to continue to deepen. He et al.<sup>25</sup> proposed a joint multi-event extraction model (JMEE) based on graph attention information aggregation. This model utilizes a multi-head attention mechanism to optimize the original GCN and improve the performance of event detection. Huang et al.<sup>26</sup> proposed a dynamic prefix method and event extraction related retrieval framework (GREE) based on generated templates. This framework can learn the specific prefix of each context and automatically identify related event types, opening up new ideas in the field of event extraction research. Wang et al.<sup>27</sup> enhanced GAT for node feature aggregation using node information and significantly improved the performance of event extraction using three-channel features. Wu et al.<sup>28</sup> designed a GAT with skip connections as a way to mine dependency trees for multi-order relationships, and then fused word contextual information and dependencies for event detection. The work of Liu et al.<sup>29</sup> designed a problem template for the event

detection task, thus redefining event detection as a machine reading comprehension (MRC) task. Huang et al.<sup>30</sup> constructed a multi-graph event extraction framework by considering the arguments as different roles, thereby solving the argument multiplexing problem. In Liu et al.<sup>31</sup>'s work, a heterogeneous graph was constructed to capture long-range dependencies between cross-sentence event arguments, and multi-granularity information was utilized to improve the accuracy of event extraction.

However, the above-mentioned models only consider semantic or grammatical patterns as additional evidence to solve the event extraction task, but rarely utilize the structured knowledge of the event itself. Moreover, they target fuzzy and unknown trigger words and the presence of noise in the text, which has not been well solved.

## Methodology

In this section, we give an overview of DAT-GAT and the proposed model is demonstrated in detail.

### Model overview

Existing event extraction models mainly utilize dynamic matching strategies to calculate cosine scores of input sentences, trigger words, and arguments in the training set. They perform dynamic matching of event patterns and use GCN to aggregate event structure and text semantic information for event extraction. However, using cosine scores as node weights only cannot effectively distinguish the importance of different event patterns and cannot solve the existing noise problem. Furthermore, GCN aggregates the information of surrounding nodes equally and cannot effectively distinguish the importance of different nodes, which adversely affects the final event extraction performance.

The event extraction model based on dynamic attention matching and GAT is divided into four parts: (1) Event pattern graph construction. This part models event structure knowledge based on words, entities, and their corresponding event annotations in the training set and constructs a global map of event patterns. (2) Dynamic attention matching. The top-k event patterns that best match the text are extracted through the dynamic attention matching mechanism, and the corresponding event pattern subgraphs are constructed. Attention scores of different event patterns are calculated. (3) Graph Attention Network (GAT). This part utilizes a graph attention network that combines event structure features to aggregate feature embeddings of node neighbors and event structure and text semantic information. (4) Joint event extraction. This part predicts trigger word types and argument roles using a trigger word classifier and argument classifier based on the self-attention mechanism. The specific model framework is illustrated in Fig. 1.

### Event pattern graph construction

To construct an event pattern graph, the text information in the training set is labeled with trigger words and arguments. Word nodes and event nodes corresponding to the trigger words and arguments are generated based on the labels. Figure 2 illustrates the construction process of the event pattern graph for the text statement “too many dispatcher configurations”. The upper part illustrates the word nodes in the sentence, and the lower part shows the event nodes generated by the annotation.

### Dynamic attention matching

The dynamic attention matching process aims to match entities in the input text information with highly similar related event nodes in the event pattern graph. This process extracts the top-k event patterns and constructs the event pattern subgraph of the corresponding sentence based on a single input sentence text. The attention mechanism assigns corresponding attention weights to each event pattern. Dynamic attention matching combines the contextual content of event trigger words to improve the accuracy of trigger word classification. For subsequent dynamic matching, ELMo<sup>8</sup> is used to encode the words in the inference sentence and the word nodes in the event pattern graph.

#### Dynamic matching of trigger words

To determine if the potential trigger word  $\rho_{t_i}$  in the input sentence  $s_t$  exists in the event pattern graph, the cosine embedding score is calculated. This score represents the semantic similarity between a potential trigger word and each trigger word node:

$$\text{sim}(\rho_{t_i}, t_{n_j}) = \cos \text{ine}(E(\rho_{t_i}), E(t_{n_j})) \quad (1)$$

where  $E$  is the ELMo embedding, and  $t_{n_j}$  is the  $j$ -th trigger text node in the information model graph.

If  $\rho_{t_i}$  does not appear as a trigger word in the training set, not only is its cosine score calculated with each trigger word node in the event pattern graph, but also the cosine score of the entire sentence's average embedding with the trigger word and the sentence embedding of the vector event pattern graph word node is calculated. The weight between the two is computed as the score to judge the semantic similarity:

$$\begin{aligned} \text{sim}_{\text{trg}}(\rho_{t_i}, t_{n_j}) = & \alpha * \cos \text{ine}(E(\rho_{t_i}), E(t_{n_j})) \\ & + (1 - \alpha) * \cos \text{ine}(E(s_t), E(t_n)) \end{aligned} \quad (2)$$

where  $\alpha$  represents the manually assigned weight, while  $E(s_t)$  and  $E(t_n)$  represent the sentence embedding of the input sentence and the context sentence of the text node, respectively.

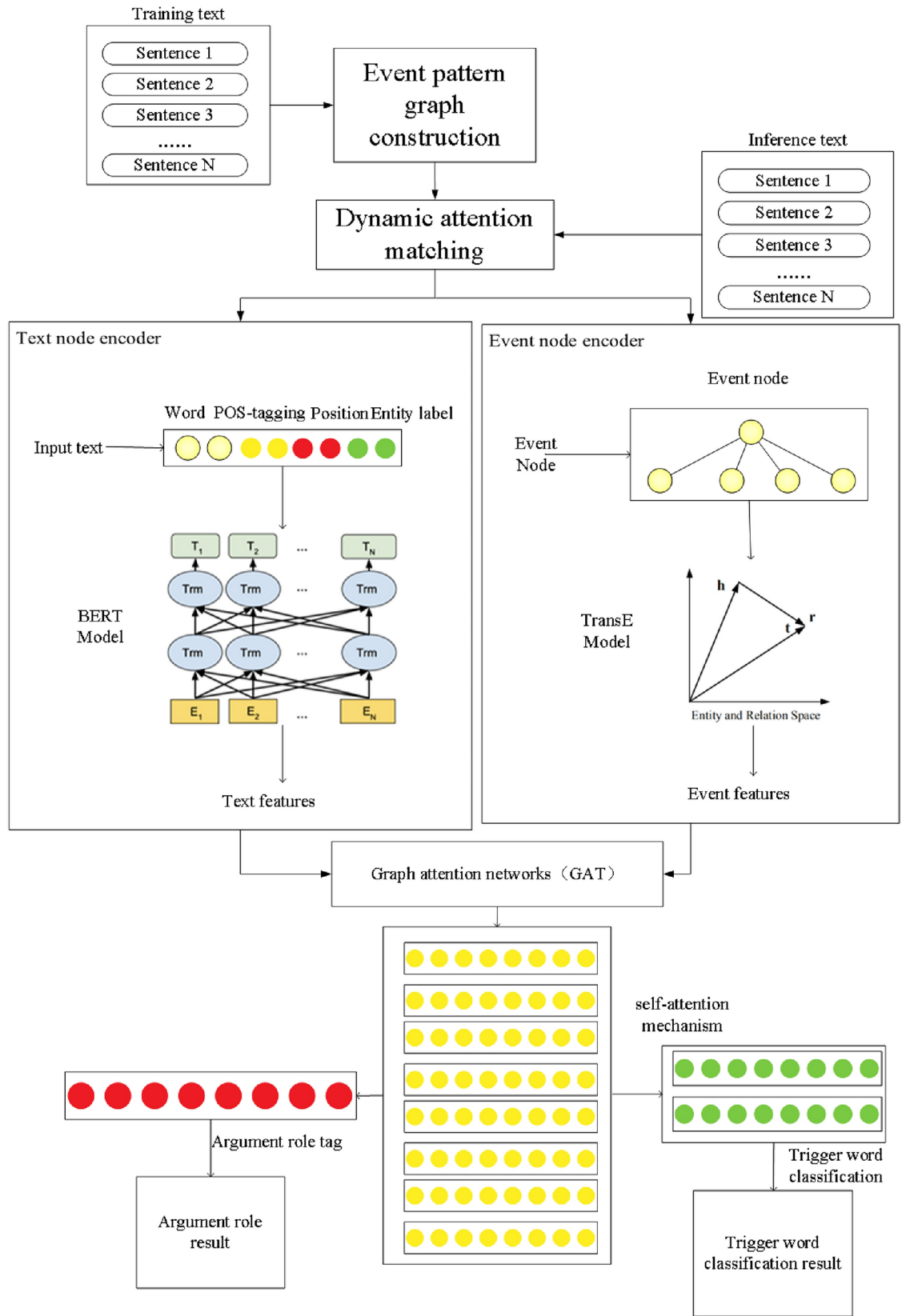


Fig. 1. Model overview.

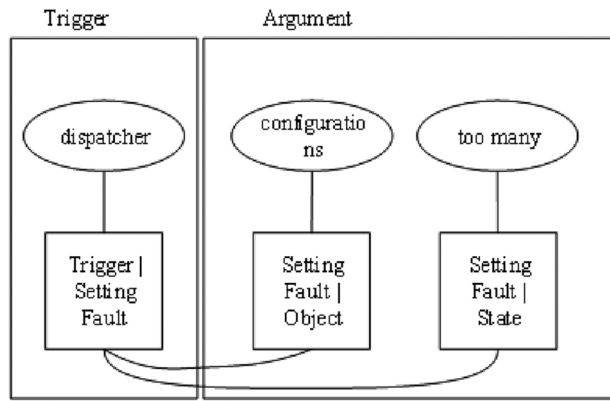


Fig. 2. Event pattern graph construction.

After comparison, the top-k matching with the highest score is selected as a candidate trigger word, and its matching trigger word type is marked as a candidate event type of the input sentence.

*Dynamic matching of argument role*

In the context of the input sentence  $s_t$ , it is plausible that each entity referenced therein can serve as an argument pertaining to the trigger word. Therefore, the proposed model is designed to dynamically align all entities with their respective arguments. To illustrate this process, let  $a_1 \dots a_m$  denote all the entities mentioned in sentence  $s_t$ , and let them be linked to the candidate trigger word  $\rho_{t_i}$ :

$$E(a_k) = \frac{1}{|a_k|} \sum_{\rho \in a_k} E(\rho) \tag{3}$$

$$E(b_k) = [E(\rho_{t_i}), E(a_k)] \tag{4}$$

where  $a_k$  denotes the k-th entity present in sentence  $s_t$  and the corresponding token in the input sentence is represented by  $\rho$ . Meanwhile,  $E(b_k)$  serves as the argument representation of  $a_k$ . In order to establish event pattern graph, the trigger word nodes and argument nodes are connected, and the cosine score is calculated by referring to the argument representation  $E(b_k)$  in sentence  $s_t$ :

$$E(b_{n_q}) = [E(t_{n_j}), E(a_{n_q})] \tag{5}$$

$$sim(a_k, a_{n_q}) = cosine(E(b_k), E(b_{n_q})) \tag{6}$$

where  $t_{n_j}$  denotes the j-th trigger word node, while  $a_{n_q}$  represents the q-th parameter word node. Additionally,  $E(b_{n_q})$  is considered as the argument representation of  $a_{n_q}$ . By comparing the scores, the top-k argument nodes are selected and subsequently matched with the argument event nodes that pertain to the candidate event type, while discarding any irrelevant nodes. Finally, the corresponding parameter event node is connected to the corresponding word node.

*Attention mechanism*

The attention mechanism is a computational approach that emulates the selective focus of human attention, and has found wide-ranging applications in feature extraction and other fields. In the present study, we employ the attention mechanism to compute the attention value between each word in the text and the candidate trigger words. The resulting event pattern vector is subjected to a nonlinear transformation layer and a softmax function, yielding an output that serves as input for the first layer of GAT aggregation weight. This enables each event pattern to be accorded a distinct level of importance. The corresponding formula is given below:

$$T = contact(E(\rho_{t_i}), E(t_{n_j})) \tag{7}$$

$$Q = W^q \cdot T \tag{8}$$

$$K = W^k \cdot T \tag{9}$$

$$V = W^v \cdot T \tag{10}$$

$$g_{(i,j)}^0 = softmax(tanh(W_i \cdot (softmax(K^T \cdot Q) \cdot V) + b)) \tag{11}$$

where  $Q, K$  and  $V$  are three eigenvector matrices. *contact* is a link function.

### Graph attention network combining event structure features

To encode input sentence information into matrix vectors, we employ the BERT model. Prior to inputting data into the BERT model, the initial matrix vector  $X$  is obtained by concatenating the word embedding, POS tag embedding, position embedding, and entity type embedding of each token in the input sentence. The resulting matrix vector is then fed into the graph attention network as a word node. In the event node encoding process, we adopt the TransE model, wherein an annotated event node is represented by  $N = \{n_1, \dots, n_t\} \in Z^{d_k \times t}$  (where  $d_k$  denotes the vector dimension set by TransE, and  $t$  represents the number of knowledge nodes). As depicted in Fig. 3, to learn a new vector representation of node  $h_i$ , the feature vectors of all its neighbors are aggregated, with the weight score of each neighbor feature being represented by  $g_{(i,j)}$ . The weight  $g_{(i,j)}^0$  that connects the initial event node and the text node is determined using the dynamic attention matching mechanism, while the weight score between nodes of the same type is set to 1. The specific process of calculating the weight score for each layer of the graph attention network is as follows:

$$g_{(i,j)} = a(W_1 \vec{h}_i, W_1 \vec{f}_j) \quad (12)$$

where  $W_1$  and  $a$  represent the learnable weight matrix and the single-layer feedforward feedback network respectively.

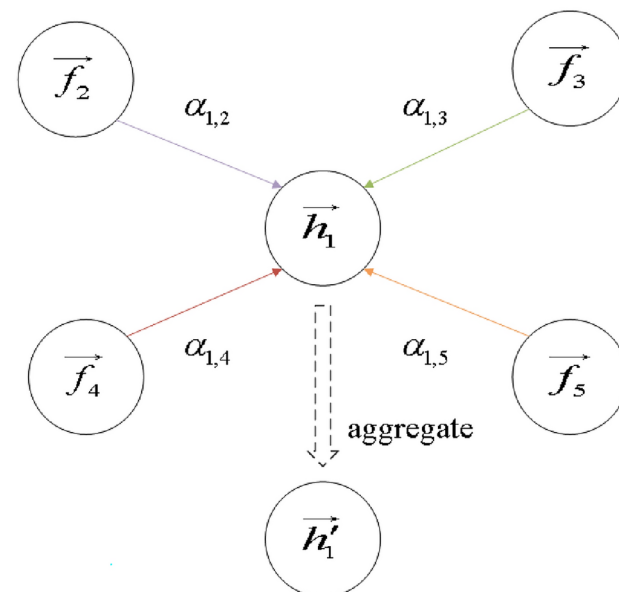
To more effectively integrate event pattern structural features, the trigger word feature vector is incorporated into the calculation of the graph attention network as a structural feature in dynamic attention matching. The weight score for each layer of the updated graph attention network is calculated as follows:

$$g_{(i,j)} = a(\mathbf{W}_1[h_i, \vec{f}_j] \odot \mathbf{W}_2[E(i), E(j)]) \quad (13)$$

where  $\odot$  represents the connection operation of matrices,  $E(i)$  represents the ELMO embedding of word  $i$ . Once the weight scores of the graph attention network are obtained, along with the structural characteristics of the event pattern, the Softmax function is employed to normalize the weights of all neighboring nodes, as illustrated below:

$$\alpha_{i,j} = \frac{\exp(g_{(i,j)})}{\sum_{\mathcal{N}_i} \exp(g_{(i,t)})} \quad (14)$$

Having obtained the attention weights of all neighboring nodes, the feature vectors of the corresponding neighbor nodes are multiplied by these weights. The resulting products are accumulated after undergoing a



**Fig. 3.** Graph attention mechanism.

matrix transformation, yielding the eigenvector of the new central node. The specific aggregation formula is given below:

$$\vec{h}_i = \sigma \left( \sum_{t \in \mathcal{N}_i} W_a \alpha_{i,t} \vec{f}_t \right) \quad (15)$$

where  $\alpha$  is the activation function and  $W_a$  is the trainable matrix.

The specific calculation process of the graph attention network described above combined with the structural characteristics of the event pattern can be expressed as:

$$\vec{h}_i^{l+1} = \sigma \left( \sum_{t \in \mathcal{N}_i} W_a^l \alpha_{i,t}^l \vec{f}_t^l \right) \quad (16)$$

## Joint extraction

### Self-attention trigger word classification

Following the GAT calculation, we obtain the feature representation of each word in the input sentence. In previous event extraction methods, there was no dynamic attention matching mechanism utilized to annotate the text before inputting it into the model. As a result, various optimization methods were employed to aggregate information at each location. In our proposed model, we leverage both the self-attention mechanism and the aforementioned dynamic attention matching mechanism to label the input text, aggregate the feature vectors of trigger words and arguments, and utilize contextual features to enhance the accuracy of trigger word classification. Through the feature representation  $S$  of the current word  $\rho_i$ , we can calculate the self-attention score vector and context vector of the position  $i$  of the current word  $\rho_i$ . The specific calculation process is as follows:

$$score = norm(\exp(W_2 f(W_1 S + b_1) + b_2)) \quad (17)$$

$$C_i = \left[ \sum_{j=1, j \neq i}^n score_j * D_j, D_i \right] \quad (18)$$

where *norm* indicates standardized operations.

The context vector  $C_i$  is subsequently input into a fully connected network, which predicts the trigger word type labels in the BIO annotation pattern, as described below:

$$\bar{C}_i = f(W_c C_i + b_c) \quad (19)$$

$$y_{t_i} = softmax(W_t \bar{C}_i + b_t) \quad (20)$$

where  $f$  is the nonlinear activation, and  $y_{t_i}$  is the final output of the  $i$ -th trigger tag.

### Argument classification

Once the correct trigger word type has been identified, the vector  $\bar{C}_i$  computed via the self-attention mechanism is combined with contextual information to classify the entity descriptions in the sentence. Given that both entities and trigger word candidates may be subsequences of tags in the text, we employ average pooling to aggregate the context vectors of the corresponding subsequences along the sequence length dimension. This yields trigger word candidate vectors  $T_i$  and entity vectors  $E_j$ , which are concatenated and input into a fully connected network to predict argument roles:

$$y_{a_{ij}} = softmax(W_a [T_i, E_j] + b_a) \quad (21)$$

where  $y_{a_{ij}}$  is the final output of the argument role played by the  $j$ -th entity in the event of the  $i$ -th trigger word candidate.

In order to better train the network, the likelihood loss function that minimizes the joint negative logarithm is used:

$$J(\theta) = - \sum_{p=1}^N \left( \sum_{i=1}^{n_p} I(y_{t_i}) \log(p(y_{t_i} | \theta)) \right) + \beta \sum_{i=1}^{t_p} \sum_{j=1}^{e_p} \log(p(y_{a_{ij}} | \theta)) \quad (22)$$

where  $N$  is the number of sentences in the training corpus;  $n_p$ ,  $t_p$  and  $e_p$  are the number of tags, the number of extracted trigger word candidates and the entity of the  $p$ -th sentence;  $I(y_{t_i})$  is an indicator function. If  $y_{t_i}$  is not the corresponding tag, it outputs a fixed The positive floating number  $\alpha$  is greater than 1, otherwise 1 is output;  $\beta$  is also a floating number used as hyperparameters such as  $\alpha$ .

## Experiments

### Datasets and evaluation metric

We also conducted experiments on a joint event extraction model based on dynamic attention matching and GAT, using the ACE2005 event extraction dataset, which comprises 599 documents containing rich text content and event information. To ensure fairness and comparability of experimental results, we followed the data segmentation method utilized in previous studies. Specifically, 529 documents were used as a training set to learn the parameters and features of the model, while 30 documents served as a development set to adjust the model hyperparameters and verify model performance. The remaining 40 documents constituted the test set, which was utilized to evaluate the generalization ability of the model. To process event types, we adopted the 33 event subtypes defined by ACE, and added the None type to cover all possible situations. In order to annotate event information in greater detail, we employed the BIO annotation mode to classify each tag into 67 different categories. This annotation method enabled accurate localization of event trigger words and argument roles, thereby providing strong support for subsequent model training.

When selecting evaluation indicators, we have opted for precision (P), recall (R), and F-measure ( $F_1$ ) as the primary indicators. These measures ensure a comprehensive reflection of the model's performance in event detection and relationship extraction tasks, thus providing an objective and quantitative basis for evaluation.

Our experimental design and evaluation standards are rigorous, ensuring the reliability and validity of the experimental results. These results provide a solid foundation for subsequent analysis and optimization of the model.

### Parameters

Our hardware configuration comprises of an Intel(R) Xeon(R) Gold 5218 CPU, NVIDIA GeForce RTX 2080Ti GPU, and 64GB RAM. The software environment includes Pycharm community 2019.2 and Pytorch 1.8.0. In the event extraction model, which is based on dynamic attention matching and GAT, we have adjusted hyperparameters on the validation set using random search. During dynamic attention matching, we select the first  $k$  parameters of the trigger word and parameter element, which is set between 0.5, ..., 1.0 and used for calculating the score of the unknown trigger word. For the word node encoder, we use 300-dimensional vectors for word embeddings and 50-dimensional vectors for the remaining three embeddings (post-embedding, position embedding, and entity type embedding). The event node encoder represents event nodes by embedding them in 440 dimensions.

### Baselines

We have compared the performance of the event extraction model based on dynamic attention matching and GAT with the following state-of-the-art event extraction models:

- *JointBeam*<sup>11</sup>, which uses a structure prediction method to extract events based on manually designed feature representation;
- *DMCNN*<sup>15</sup>, which utilizes dynamic multi-pooling operations to preserve information of multiple events;
- *JRNN*<sup>32</sup>, which uses bidirectional RNN and manually designed features to jointly extract event triggers and parameters;
- *dbRNN*<sup>33</sup>, which incorporates a dependency bridge on Bi-LSTM for event extraction;
- *JMEE*<sup>19</sup>, which employs attention-based GCN to construct the corresponding syntactic tree through sentences as the path of the semantic graph to aggregate word node information;
- *GREE*<sup>26</sup>, which introduces a method based on generating templates, an event extraction framework with dynamic prefixes, and correlation retrieval. The specific experimental results are presented in Table 1:

Table 1 presents the event extraction results on the ACE2005 dataset, where our model achieves good  $F_1$  scores in both trigger word classification and argument-related subtasks. Notably, our model achieves the best performance in trigger word classification after incorporating event structure knowledge. Our model achieves the highest precision, recall, and  $F_1$  scores among all selected joint extraction models. Compared with the currently best-performing reported models, our model shows improvements of 2.4%, 1.2%, and 2.3%, respectively. The overall performance in argument identification and argument role-related tasks also shows a slight improvement. Although the performance in trigger word identification is not significantly better than the best reported model, the  $F_1$  score is only 0.1% lower than the best-performing JMEE model. This is attributed to our proposed dynamic attentional matching, which identifies sufficient information about the event structure and uses attention mechanisms to better filter the noise of the event pattern. Furthermore in comparison with the GCN-based model JMEE, the overall performance in the tasks related to trigger classification, argument recognition and argument roles is also improved, the  $F_1$  scores are improved by 2.1%, 0.4%, 0.7%, respectively, which demonstrates that the importance of different nodes in the event graph can be efficiently differentiated through the introduction of the graph attention mechanism, thus improving the model performance.

Approach	Trigger Identification (%)			Trigger Classification (%)			Argument Identification (%)			Argument Role(%)		
	P	R	$F_1$	P	R	$F_1$	P	R	$F_1$	P	R	$F_1$
JointBeam	74.7	62.8	68.2	71.4	60.3	65.4	66.9	45.2	53.9	61.2	41.5	49.5
DMCNN	<b>78.8</b>	65.1	71.3	73.5	61.2	66.8	65.8	48.7	56.0	59.9	43.1	50.1
JRNN	66.3	<b>72.9</b>	69.4	64.1	71.0	67.4	58.7	61.4	60.0	51.5	54.3	52.9
dbRNN	N/A			71.7	67.4	69.5	69.7	63.0	66.2	64.1	50.1	56.2
JMEE	78.5	70.1	<b>74.1</b>	73.9	70.1	71.9	69.7	63.8	66.6	<i>64.5</i>	52.1	57.6
GREE	N/A			68.6	71.2	69.9	68.2	63.1	65.6	64.0	51.1	56.8
Our model	77.8	70.5	74.0	<b>76.3</b>	<b>72.2</b>	<b>74.2</b>	<b>70.2</b>	<b>64.0</b>	<b>67.0</b>	<b>65.2</b>	<b>52.8</b>	<b>58.3</b>

**Table 1.** Experimental results on the ACE2005 data set (%). The best results are in bold and the second best results are in italics.

Approach	Trigger Identification (%)			Trigger Classification (%)			Argument Identification (%)			Argument Role(%)		
	P	R	$F_1$	P	R	$F_1$	P	R	$F_1$	P	R	$F_1$
JMEE	<b>78.5</b>	70.1	<b>74.1</b>	73.9	70.1	71.9	69.7	63.8	66.6	<i>64.5</i>	52.1	57.6
DAM-JMEE	77.3	70.2	73.6	75.9	71.3	73.5	69.9	<b>64.2</b>	66.9	64.8	52.3	57.9
GAT	77.4	70.4	73.7	75.1	71.9	73.5	<b>70.5</b>	63.5	66.8	<i>65.1</i>	52.7	58.2
DAM-GAT	77.8	<b>70.5</b>	<b>76.3</b>	<b>72.2</b>	<b>74.2</b>	70.2	64.0	<b>67.0</b>	<b>65.2</b>	<b>52.8</b>	<b>58.3</b>	

**Table 2.** Ablation studies on ACE2005 dataset. The best results are in bold and the second best results are in italics.

### Ablation studies

This study investigates the effectiveness of two modules, Dynamic Attention Matching (DAM) and Graph Attention Network (GAT), in enhancing joint event extraction performance by combining event structural features. We evaluate the performance of the model combining different modules on the ACE2005 dataset, and the results are presented in Table 2. To demonstrate the superiority of GAT in the proposed model on joint event extraction performance, we use the baseline model JMEE as a comparison and label it as JMEE. We evaluate the impact of JMEE and DAM-JMEE on joint event extraction performance. Furthermore, we remove the dynamic attention matching module from the model to evaluate its impact on joint event extraction performance, denoted as GAT and JMEE.

Our comparison of P, R, and  $F_1$  scores between JMEE and DAM-JMEE, as well as GAT and DAM-GAT, indicates that dynamic attention matching (DAM) has a significant impact on event extraction performance, particularly in trigger word classification tasks and argument identification and classification tasks. Although the performance improvement of DAM-JMEE compared to the JMEE model is not significant in the trigger word recognition task, the superiority of the dynamic attention matching module is clearly visible in the comparison between GAT and DAM-GAT. Similarly, by comparing the P, R, and  $F_1$  scores of JMEE and GAT, as well as DAM-JMEE and DAM-GAT, we observe that the graph attention network (GAT) combined with event structural features improves event extraction trigger word classification tasks and argument recognition performance in classification tasks. Although the performance improvement of JMEE compared to the GAT model is not significant in the trigger word recognition task, the superiority of the graph attention network module that incorporates event structural features is evident in the comparison between DAM-JMEE and DAM-GAT. In summary, our ablation experimental results clearly demonstrate the effectiveness of dynamic attention matching and graph attention networks incorporating event structural features in enhancing event extraction performance.

### Conclusion

This paper proposes a joint event extraction model that utilizes a dynamic attention matching mechanism and GAT to aggregate event structure and text semantic information, thereby improving the recognition and classification performance of trigger words and arguments. The model first extracts the top-k event patterns that best match the text through dynamic attention matching and constructs corresponding event pattern subgraphs while calculating attention scores. Next, the graph attention network aggregates the node neighbor characteristics, embedding and aggregating event structure and text semantic information. Finally, the self-attention mechanism-based trigger word and argument classifiers predict the trigger word type and argument role. Experimental results demonstrate that the proposed model outperforms existing models in event extraction.

It is worth noting that the event pattern graph may contain too many trigger words and argument nodes. To address this issue, the model initially performs dynamic matching to extract the top-k event pattern types and then uses the attention mechanism to allocate weights. Although this approach is not perfect, directly using

the attention mechanism for dynamic matching calculations would require significant computing power and time costs. This is one of the areas we will consider for model optimization in the future. Additionally, we plan to integrate multimodal data and event knowledge graphs in our future research, providing new research ideas for interdisciplinary collaboration and innovation in the fields of natural language processing, computer vision, knowledge representation, and machine learning.

### Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 15 October 2024; Accepted: 20 February 2025

Published online: 26 February 2025

### References

- Li, M. et al. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7–11 November, 2021* (eds Moens, M. et al.) 5203–5215 (Association for Computational Linguistics, 2021). <https://doi.org/10.18653/V1/2021.EMNLP-MAIN.422>.
- Naik, A. & Rosé, C. P. Towards open domain event trigger identification using adversarial domain adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020* (eds Jurafsky, D. et al.) 7618–7624 (Association for Computational Linguistics, 2020). <https://doi.org/10.18653/V1/2020.ACL-MAIN.681>.
- Wang, R., Li, B., Hu, S., Du, W. & Zhang, M. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access* **8**, 5212–5224. <https://doi.org/10.1109/ACCESS.2019.2963367> (2020).
- Costa, T. S., Gottschalk, S. & Demidova, E. Event-qa: A dataset for event-centric question answering over knowledge graphs. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020* (eds d'Aquin, M. et al.) 3157–3164 (ACM, 2020). <https://doi.org/10.1145/3340531.3412760>.
- Du, X., Rush, A. M. & Cardie, C. GRIT: generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19–23, 2021* (eds Merlo, P. et al.) 634–644 (Association for Computational Linguistics, 2021). <https://doi.org/10.18653/V1/2021.EACL-MAIN.52>.
- Wan, Q., Wan, C., Hu, R. & Liu, D. Chinese financial event extraction based on syntactic and semantic dependency parsing. *Chin. J. Comput.* **44**, 508–530 (2021).
- Song, Y. Construction of event knowledge graph based on semantic analysis. *Tehniki vjesnik* **28**, 1640–1646 (2021).
- Chen, Y. et al. A history and theory of textual event detection and recognition. *IEEE Access* **8**, 201371–201392. <https://doi.org/10.1109/ACCESS.2020.3034907> (2020).
- Du, L., Ding, X., Liu, T. & Qin, B. Learning event graph knowledge for abductive reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021* (eds Zong, C. et al.) 5181–5190 (Association for Computational Linguistics, 2021). <https://doi.org/10.18653/V1/2021.ACL-LONG.403>.
- Lybarger, K., Ostendorf, M., Thompson, M. & Yetisgen, M. Extracting COVID-19 diagnoses and symptoms from clinical text: A new annotated corpus and neural event extraction framework. *J. Biomed. Informatics* **117**, 103761. <https://doi.org/10.1016/J.JBI.2021.103761> (2021).
- Li, Q., Ji, H. & Huang, L. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers* (ed. Li, Q.) 73–82 (The Association for Computer Linguistics, 2013).
- Chen, Y., Li, W., Liu, Y., Zheng, D. & Zhao, T. Exploring deep belief network for chinese relation extraction. In: *CIPS-SIGHAN Joint Conference on Chinese Language Processing* (2010).
- Hong, Y. et al. Using cross-entity inference to improve event extraction. In: *Proc. 49th annual meeting of the association for computational linguistics: human language technologies*. 1127–1136 (2011).
- Venugopal, D., Chen, C., Gogate, V. & Ng, V. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In: *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 831–843 (2014).
- Chen, Y., Xu, L., Liu, K., Zeng, D. & Zhao, J. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers* 167–176 (The Association for Computer Linguistics, 2015). <https://doi.org/10.3115/V1/P15-1017>.
- Nguyen, T. & Grishman, R. Graph convolutional networks with argument-aware pooling for event detection. In: *Proc. AAAI Conference on Artificial Intelligence*, vol. 32 (2018).
- Lu, Y., Lin, H., Han, X. & Sun, L. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In: *Proc. 57th Annual Meeting of the Association for Computational Linguistics*, 4366–4376 (2019).
- Zhang, Y. et al. Event detection with dynamic word-trigger-argument graph neural networks. *IEEE Trans. Knowl. Data Eng.* **35**, 3858–3869. <https://doi.org/10.1109/TKDE.2021.3132956> (2023).
- Liu, X., Luo, Z. & Huang, H. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (eds Riloff, E. et al.) 1247–1256 (Association for Computational Linguistics, 2018). <https://doi.org/10.18653/V1/D18-1156>.
- Wadden, D., Wennberg, U., Luan, Y. & Hajishirzi, H. Entity, relation, and event extraction with contextualized span representations. 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 2019, Hong Kong, China, November 3–7, 2019* (eds Inui, K. et al.) 5783–5788 (Association for Computational Linguistics, 2019). <https://doi.org/10.18653/V1/D19-1585>.
- Lin, Y., Ji, H., Huang, F. & Wu, L. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020* (eds Jurafsky, D. et al.) 7999–8009 (Association for Computational Linguistics, 2020). <https://doi.org/10.18653/V1/2020.ACL-MAIN.713>.
- Du, X. & Cardie, C. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020* (eds Webber, B. et al.) 671–683 (Association for Computational Linguistics, 2020). <https://doi.org/10.18653/V1/2020.EMNLP-MAIN.49>.
- Su, F., Zhang, Y., Li, F. & Ji, D. Balancing precision and recall for neural biomedical event extraction. *IEEE ACM Trans. Audio Speech Lang. Process.* **30**, 1637–1649. <https://doi.org/10.1109/TASLP.2022.3161146> (2022).
- Li, Q. et al. Reinforcement learning-based dialogue guided event extraction to exploit argument relations. *IEEE ACM Trans. Audio Speech Lang. Process.* **30**, 520–533. <https://doi.org/10.1109/TASLP.2021.3138670> (2022).

25. He, L., Meng, Q., Zhang, Q., Duan, J. & Wang, H. Event detection using a self-constructed dependency and graph convolution network. *Appl. Sci.* **13**, 3919 (2023).
26. Huang, H., Liu, X., Shi, G. & Liu, Q. Event extraction with dynamic prefix tuning and relevance retrieval. *IEEE Trans. Knowl. Data Eng.* **35**, 9946–9958. <https://doi.org/10.1109/TKDE.2023.3266495> (2023).
27. Wan, Q., Wan, C., Xiao, K., Hu, R. & Liu, D. A multi-channel hierarchical graph attention network for open event extraction. *ACM Trans. Inf. Syst.* **41**, 1–27 (2023).
28. Wu, G., Lu, Z., Zhuo, X., Bao, X. & Wu, X. Semantic fusion enhanced event detection via multi-graph attention network with skip connection. *IEEE Trans. Emerg. Top. Computat. Intell.* **7**, 931–941 (2023).
29. Liu, L., Liu, M., Liu, S. & Ding, K. Event extraction as machine reading comprehension with question-context bridging. *Knowl.-Based Syst.* 112041 (2024).
30. Huang, H. et al. A multi-graph representation for event extraction. *Artif. Intell.* **332**, 104144 (2024).
31. Liu, Y., Gao, N., Zhang, Y. & Kong, Z. Enhancing document-level event extraction via structure-aware heterogeneous graph with multi-granularity subsentences. In *ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (ed. Liu, Y.) 12657–12661 (IEEE, 2024).
32. Nguyen, T. H., Cho, K. & Grishman, R. Joint event extraction via recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016* (eds Knight, K. et al.) 300–309 (The Association for Computational Linguistics, 2016). <https://doi.org/10.18653/V1/N16-1034>.
33. Sha, L., Qian, F., Chang, B. & Sui, Z. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (eds McIlraith, S. A. & Weinberger, K. Q.) 5916–5923 (AAAI Press, 2018). <https://doi.org/10.1609/AAAI.V32I1.12034>.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (62071240), the Innovation Program for Quantum Science and Technology (2021ZD0302901), the Natural Science Foundation of Jiangsu Province (BK20231142), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

## Author contributions

J.C., W.L.: Conceptualization, Methodology, Software, Investigation, Writing- Original draft preparation. Z.W.: Data curation, Validation, Supervision, Resources, Writing - Review & Editing. Z.R., X.L.: Project administration, Supervision, Resources, Writing - Review & Editing.

## Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to W.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025