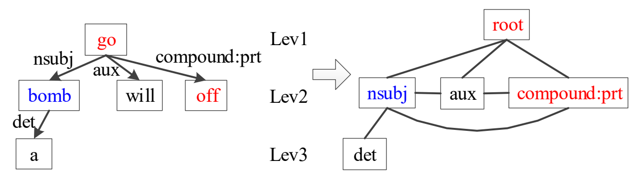
Dependency Structure-Enhanced Graph Attention Networks for Event Detection

Qizhi Wan[1,2], Changxuan Wan[1,2*], Keli Xiao[3*], Kun Lu[4], Chenliang Li[5], Xiping Liu[1,2], Dexi Liu[1,2]

[1]School of Information Management, Jiangxi University of Finance and Economics
[2]Jiangxi Key Lab of Data and Knowledge Engineering
[3]College of Business, Stony Brook University
[4]School of Library and Information Studies, The University of Oklahoma
[5]School of Cyber Science and Engineering, Wuhan University
{wanqizhi1006@163.com, wanchangxuan@263.net, keli.xiao@stonybrook.edu}

Abstract

Existing models on event detection share three-fold limitations, including (1) insufficient consideration of the structures between dependency relations, (2) limited exploration of the directed-edge semantics, and (3) issues in strengthening the event core arguments. To tackle these problems, we propose a dependency structure-enhanced event detection framework. In addition to the traditional token dependency parsing tree, denoted as TDG, our model considers the dependency edges in it as new nodes and constructs a dependency relation graph (DRG). DRG allows the embedding representations of dependency relations to be updated as nodes rather than edges in a graph neural network. Moreover, the levels of core argument nodes in the two graphs are adjusted by dependency relation types in TDG to enhance their status. Subsequently, the two graphs are further encoded and jointly trained in graph attention networks (GAT). Importantly, we design an interaction strategy of node embedding for the two graphs and refine the attention coefficient computational method to encode the semantic meaning of directed edges. Extensive experiments are conducted to validate the effectiveness of our method, and the results confirm its superiority over the state-of-the-art baselines. Our model outperforms the best benchmark with the $F1$ score increased by 3.5 and 3.4 percentage points on ACE2005 English and Chinese corpus.



(a) TDG (b) DRG

Figure 1: Illustration of the *Attack* event type from ACE2005 (sentence example: "A bomb will go off"). (a) displays the token dependency tree (TDG), where triggers annotated in the corpus are highlighted in red, and the core arguments, as per our definition, are in blue. The dependency relation types shown include *det*, *nsubj*, *aux*, and *compound:prt*. The *root* type is the dependency relation of "go" and its parent ROOT node (a virtual node). (b) presents the corresponding dependency relation graph (DRG) constructed in this paper. Here, a node, termed a "dependency relation node," corresponds to a dependency edge in the TDG. If two TDG dependency edges share a token node, an edge is established between the dependency relation nodes in the DRG. Lev$X$ indicates the X level in the graph structures.

Introduction

Event detection is an essential and challenging task of information extraction. Its target is to identify the events and their types in texts, and a common implementation is to detect whether each token in the sentence triggers a specific event type.

Token dependency information, mainly including the token syntactic dependency structure and dependency relations (i.e., relation types), is usually demonstrated in the form of a token dependency tree/graph (TDG) by the dependency parsing tool. Most recent studies on event detection utilized graph neural networks for encoding token dependency information, showing the effectiveness of TDG. Based on TDG, Cui et al. (2020) updated edge representations to obtain dependency relation information, and Wan et al. (2023a) used static dependency relation representations to compute edge attentions. Liu, Xu, and Liu (2021) devised a self-attention graph residual convolution network to mine node-to-node latent dependency relations. Mi, Hu, and Li (2022) performed event detection with dual relational graph attention networks based on two graphs constructed by dependency parsing and semantic role labeling. Despite TDG being widely studied, the dependency information can be further exploited for better implementations, which we summarize into the following three aspects.

First, the structures between dependency relations have not been sufficiently explored. Merely using the dependency relations based on TDG (Cui et al. 2020; Wan et al. 2023a), some important information regarding associations between different dependency relations may be ignored. The TDG merely describes the dependency relations between tokens, but not the structures between dependency relations. For example, Figure 1(a) demonstrates a token dependency tree. Existing work only considers the token dependency in-

* Corresponding Author.

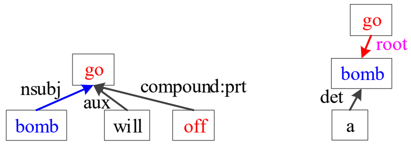


Figure 2: Demonstrating node embedding aggregation based on TDG in Figure 1. (a) and (b) illustrate the aggregation of nodes "go" and "bomb", respectively. In (a), the edge weight from "bomb" to "go" (blue line) is calculated using the embeddings of the token "bomb" and the dependency relation *nsubj*. In (b), the edge weight for "go" pointing to "bomb" (red line) is determined by the embeddings of the token "go" and the dependency relation *root*.

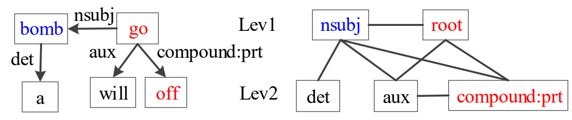


Figure 3: The final structures after core argument adjustment based on Figure 1. The "bomb" and *nsubj* nodes are promoted from level 2 to level 1.

formation in it and neglects the structure information between dependency relations, which can be constructed as a *Dependency Relation Graph* (DRG) in Figure 1(b).

Constructing DRG based on TDG brings the following two benefits. On the one hand, dependency relation (i.e., edge) representations can be aggregated by the graph neural network mechanism on DRG in which the dependency relations are converted to nodes. Although they can also be updated on TDG, the implementation is generally done by a dot product of nodes and edges, and the structure information among the edges is ignored (Cui et al. 2020). Moreover, the representations of tokens and dependency relations can be interactively updated on TDG and DRG with different structures. For instance, the representations of nodes in DRG will influence the edge weights in TDG when implementing the node aggregation based on TDG, and the reverse is the same. On the other hand, the performance degradation issue caused by stacking multiple layers of GNN (i.e., over-smoothing) (Yan et al. 2019) can be alleviated. For example, in Figure 1(a), the information of the *nsubj* edge can only be delivered to the *compound:prt* edge through the path *nsubj*→bomb→go→off→*compound:prt*. Here, tokens are needed as bridges between them, while they can be directly propagated in the DRG of Figure 1(b), greatly reducing the depth of GNN layers.

Second, semantic meanings of different directions of edges in TDG are not distinguished. Regarding edge attention coefficients, previous studies (Cui et al. 2020; Wan et al. 2023a) failed to sufficiently consider the dependency relation types and the direction of edges in TDG, by simply computing the edge attention coefficient based on the representations of the current node and its neighbor nodes. For example, as shown in Figure 2(a), when implementing the aggregation for node "go", the dependency relation *nsubj* (nominal subject) between "bomb" and "go" should be used because "bomb" depends on "go" by the nominal subject role. However, in Figure 2(b), if the aggregated object changes to the "bomb" node, this dependency relation is not appropriate for computing the edge attention coefficient, given that "go" does not depend on "bomb". To reveal the dependency role of "go", we should consider the dependency relation (i.e., *root*) between "go" and its parent as the relation type of "go" aggregating to "bomb". Along this line, we can capture the semantics of the neighbor's role in the sentence.

Finally, the event type cannot be easily identified only by the trigger. As shown in Figure 1(a), it is difficult to determine the event type as *Attack* by the trigger "go". For example, the trigger of the sentence "we go up there" is also "go", while the event type is *Transport*. Intuitively, the argument "bomb" is a pivotal token to determine the event type. However, these core arguments (defined in this paper) are mainly distributed in the subject and object of an event, and their levels are lower than the trigger. To reinforce their statues, we elevate the levels of these core argument nodes in TDG and corresponding dependency relation nodes in DRG according to tokens' dependency relation types in TDG, instead of using the annotated gold arguments. The final structures of the two graphs are shown in Figure 3.

To implement the aforementioned ideas, this paper develops a Dependency Structure-Enhanced Event Detection (DSE-ED) model. Given an input sentence, the TDG and DRG are constructed in sequence. Then, the levels of core argument nodes and corresponding dependency relation nodes in the two graphs are elevated. After that, the method of calculating the edge attention coefficient of Graph Attention Network (GAT) (Velickovic et al. 2018) is refined by jointly leveraging the semantics of tokens and dependency relations. Meanwhile, the GAT model is employed to encode the two graphs' structures and other features. Importantly, node representations in the two graphs interact at each layer. Finally, the output representations of each token node and its corresponding dependency relation node at the last layer of GATs are concatenated and fed into a fully connected network for event-type recognition.

To sum up, main contributions of this work are four-fold.

- We construct the dependency relation graph (DRG), and this is the first model that considers a dependency relation edge as a new node for graph construction. Also, token and dependency relation representations can be interactively updated on the two graphs with different structures, respectively, capturing more semantic information.
- To encode the structure semantics of dependency edges in diverse directions, we use the representations of the current token and the dependency relations of its neighbors to refine the attention mechanism of GAT.
- To strengthen core argument status, we raise their levels in the two graphs. As a newly discovered clue, the process is found to be helpful in identifying the event type.

- Extensive experiments are conducted on two datasets, and the results confirm the effectiveness of our scheme in handling the event detection task by achieving an improvement of 3.5∼18.6 and 3.4∼34.1 percentage points, respectively.

Related Work

In recent years, event detection has been widely studied and achieved encouraging results (Wan et al. 2021, 2023b,c). In terms of the token dependency graph (TDG), Sha et al. (2018) added the dependency structure into an LSTM model to improve its encoding ability. To address the token long-dependency problem, Nguyen and Grishman (2018) and Liu, Luo, and Huang (2018) adopted the Graph Convolutional Network (GCN) to encode the token dependency structure. Argument pooling and self-attention were considered in their models, respectively. Yan et al. (2019) divided the token dependency structure into three substructures and used three GATs to encode the information in parallel. Subsequently, to capture more structured information, Pouran Ben Veyseh, Ngo Nguyen, and Nguyen (2020) combined the syntactic and semantic structures of tokens into a graph. Although using the GCN model, Lai, Nguyen, and Nguyen (2020) focused on filtering out the token information unrelated to candidate triggers. Cui et al. (2020) was the first to encode the dependency relation information by interacting with the embeddings of tokens and dependency relations.

Also, to enrich the semantic meanings of tokens without direct dependency correlation in TDG, Ahmad, Peng, and Chang (2021) designed a graph attention transformer encoder. Considering that the token dependency structure is unchanged in previous studies and partial edges are noise for event recognition, Liu, Xu, and Liu (2021) pruned these noise edges and proposed a graph residual convolutional network. By adding co-reference edges and adjusting parallel structures, Wan et al. (2023a) developed a bidirectional token dependency graph to handle the open event extraction.

Regarding argument utilization, Liu et al. (2017) strengthened the argument by eliminating the ambiguity of triggers, and Nguyen and Grishman (2018) explicitly aggregated the embedding information of entities in the GCN model. Sha et al. (2018) improved the argument extraction effect by establishing the associations between arguments.

Our work addresses the following limitations in existing studies. (1) There is insufficient exploitation of dependency relation structure semantics. (2) Dependency relation embeddings are updated only by token embeddings in TDG. (3) There are limited discussions on dependency edge semantics of the aggregation node, in which the weight is determined only by the token or token dependency information. (4) There is an insufficient utilization of argument importance, such as explicitly raising their levels.

Methodology

Following previous studies (Cui et al. 2020; Wan et al. 2023a), we formulate the detection task as a sequence labeling task; that is, each token is assigned a label that contributes to event annotation. The “O” refers to the “Other” tag, meaning that the corresponding token is irrelevant to the target event. Regarding the event trigger, we use the “B-*Event Type*” and “I-*Event Type*” tags to represent the position and type of a token in the event trigger, where “B” and “I” denote “begin” and “inside”, respectively.

In the following, we describe our Dependency Structure-Enhanced Event Detection framework (DSE-ED). As demonstrated in Figure 4, the framework includes six major components: (1) the **Embedding Layer**, for initializing the semantic embeddings of tokens and dependency relations in sentences; (2) the **Bi-LSTM Layer**, learning token sequential semantics and dependency relation structural semantics; (3) the **Graph Structure**, functional in constructing the token dependency graph and dependency relation graph; (4) the **Core Argument Adjustment**, for the level elevation of core arguments and corresponding dependency relation nodes; (5) the **Dependency Structure-Enhanced GAT**, encoding the token dependency and dependency relation graph by the token dependency GAT and dependency relation GAT; (6) the **Classification Layer**, for predicting the event type.

Embedding Layer

We use the BERT (Devlin et al. 2019) to initialize token embeddings. Given a sentence $s = \{w_1, \cdots, w_i, \cdots, w_n\}$, the initial vector of the i-th token w_i is denoted as $\mathbf{v}_{\mathrm{sem}_i}$, where n represents the length of s. In TDG, the dependency relation between node i and its parent node is taken as i's dependency relation. Note that, the virtual root node is not considered. The dependency relation embedding of w_i is generated by looking up the randomly initialized embedding table, denoted as $\mathbf{v}_{\mathrm{dep}_i}$.

Bi-LSTM Layer

To capture the sequential information of tokens and dependency relations, two Bi-LSTM networks (Hochreiter and Schmidhuber 1997) are employed, respectively. The output embeddings of the i-th token and dependency relation are denoted as $\mathbf{h}_{\mathrm{sem}_i}$ and $\mathbf{h}_{\mathrm{dep}_i}$.

Graph Structure

Given the sentence s, the token dependency tree is built by a syntactic parsing tool. According to the Introduction section, different directions of an edge in the tree can demonstrate various semantic meanings. Therefore, the token dependency graph (TDG) adjusted based on the tree is bidirectional. Subsequently, we consider dependency edges in the graph as new nodes (dependency relation nodes) and then establish new edges between new nodes if they share the same token nodes; thus, the dependency relation graph (DRG) is constructed. As edges in DRG reveal the situation of sharing tokens and the semantic meaning of an edge in diverse directions is the same, DRG is an undirected graph. For the head node in TDG, the dependency relation between it and the root node is also included in the construction, as shown in the red line box in Figure 4. Therefore, the node number of the two graphs is identical, and nodes can be matched one-to-one.

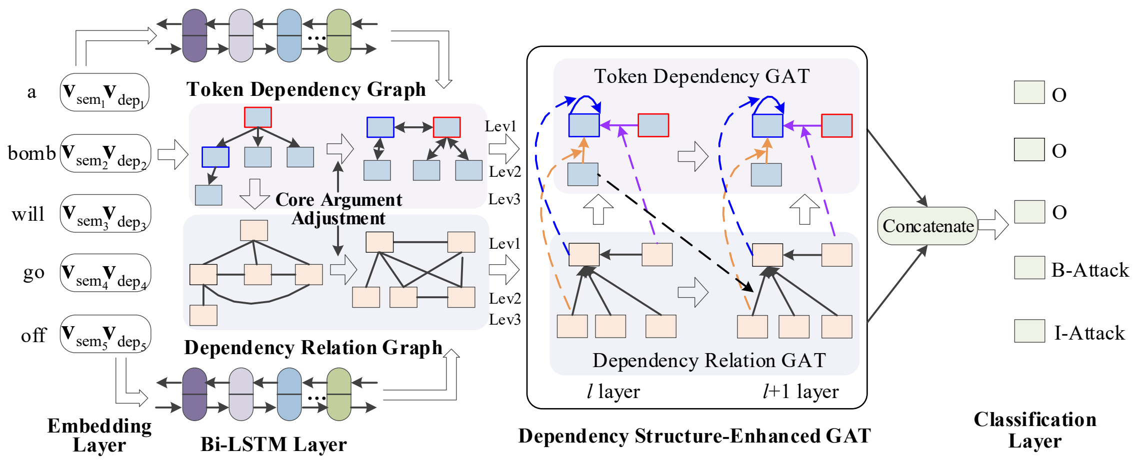


Figure 4: Dependency structure-enhanced event detection framework (DSE-ED). Lev$X$ denotes the X level in the tree/graph. Core arguments and head nodes, as parsed by the syntactic tool, are represented by blue and red line boxes, respectively. In the dependency structure-enhanced GAT module, dotted lines pointing from DRGs to TDGs indicate that the node embeddings in DRGs regulate the attention coefficients of edges in TDGs. The meaning of dotted lines pointing from TDG to DRG is similar; that is, the latest token embedding in the current layer computes the edge weight of the next layer in DRG. For visual clarity, only the aggregation of core arguments is demonstrated.

Core Argument Adjustment

Given that the majority of arguments affecting event type recognition act as the subject and object of an event, we define the arguments whose dependency relations are *subject-predicate* and *predicate-object* as core arguments. Furthermore, considering parallel tokens/clauses (the dependency relation type is parallel meaning) having the same status, this paper also raises their levels.

Specifically, if a token's dependency relation type in TDG falls into one of the three categories, the token is promoted to the same level as its parent, and the corresponding dependency relation node in DRG is promoted accordingly. Note that the dependency relation between a node and its parent is represented as the node's. For example, the blue line box node in Figure 4 is adjusted from level 2 to level 1 as its dependency relation meets the adjustment condition.

Regarding the levels of tokens and dependency relations, they are used to affect the weight when aggregating node's embedding in the graph attention network, as shown in Equation (4). The Equation (5) gives the calculation method.

Dependency Structure-Enhanced Graph Attention Network

Dependency Relation Graph Attention Network. For the original graph attention network (Velickovic et al. 2018), the input of the building block layer is a set of node embedding, $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_i, \cdots, \mathbf{h}_n\}$, where n is the number of nodes, $\mathbf{h}_i \in \mathbb{R}^D$, D is the embedding representation dimension. The layer produces a new set of node embedding (dimension is D'), $\mathbf{H}' = \{\mathbf{h}'_1, \cdots, \mathbf{h}'_i, \cdots, \mathbf{h}'_n\}$, $\mathbf{h}'_i \in \mathbb{R}^{D'}$.

The attention coefficient α_{ij} representing the importance of node j's embedding to node i can be computed as follows:

$$\alpha_{ij} = \frac{\exp\left(\phi\left(\varphi^{\mathrm{T}}\left[\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j\right]\right)\right)}{\sum_{k\in V_i}\exp\left(\phi\left(\varphi^{\mathrm{T}}\left[\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_k\right]\right)\right)}, \tag{1}$$

where ϕ is a nonlinearity activation function (LeakyReLU), φ is a weight vector, $\cdot^{\mathrm{T}}$ represents the transposition operation, $\mathbf{W} \in \mathbb{R}^{D'\times D}$ is a weight matrix, and V_i represents a set of i's neighbors in the graph.

Hence, the output embedding of node i based on the multi-head attention mechanism can be written as:

$$\mathbf{h}'_i = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in V_i}\alpha_{ij}^{m}\mathbf{W}^{m}\mathbf{h}_j\right), \tag{2}$$

where M is the number of attention mechanism head; σ is a nonlinear function; α_{ij}^m is the normalized attention coefficient computed by the m-th attention mechanism (α^m), and $\mathbf{W}^m$ is the corresponding weight matrix.

Since the edge in DRG refers to sharing the same token and the meaning cannot be differentiated, we combine the embedding of the current dependency relation node with the corresponding token embedding to control the edge weight. Therefore, based on the original attention mechanism in GAT, the attention coefficient and the dependency relation embedding of l-th layer with token level feature in the dependency relation GAT are defined as follows:

$$\gamma_{ij}^{(l)} = \frac{\exp\left(\phi\left(\varphi^{\mathrm{T}}\mathbf{W}^{(l)}\mathbf{h}_{\mathrm{sem}_j}^{(l-1)}(\mathbf{h}_{\mathrm{dep}_i}^{(l-1)})^{\mathrm{T}}\right)\right)}{\sum_{k\in V_i}\exp\left(\phi\left(\varphi^{\mathrm{T}}\mathbf{W}^{(l)}\mathbf{h}_{\mathrm{sem}_k}^{(l-1)}(\mathbf{h}_{\mathrm{dep}_i}^{(l-1)})^{\mathrm{T}}\right)\right)}, \tag{3}$$

$$\mathbf{h}_{\mathrm{dep}_i}^{(l)} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in V_i}u_j\gamma_{ij}^{m(l)}\mathbf{W}^{m(l)}\mathbf{h}_{\mathrm{dep}_j}^{(l-1)}\right), \tag{4}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{F\times D}$, F and D refer to the dimension of dependency relation and token embedding, $\mathbf{h}_{\mathrm{sem}_j}^{(l-1)} \in \mathbb{R}^D$ is the latest embedding of node j output by the l-1 layer in the token dependency GAT, $\mathbf{h}_{\mathrm{dep}_i}^{(l-1)} \in \mathbb{R}^F$, and u_j represents the normalized value of node level. Following Wan et al. (2023a), we formulate node's level as:

$$u_j = \exp(-lev_j), \tag{5}$$

where lev_j, a natural number, represents the level of node *j*.

Compared with existing work that merely focuses on the token dependency information in the token dependency tree and ignores the structures between dependency relations, the dependency relation GAT can capture these structures to obtain the essential semantics.

Token Dependency Graph Attention Network. To embody the importance of various dependency relations, the embeddings of the current token node and its neighbors' dependency relations are exploited to jointly regulate the weight of the dependency edge in the token dependency graph, distinguishing diverse directions. The attention coefficient of *l*-th layer can be computed as follows:

$$\beta_{ij}^{(l)} = \frac{\exp\left(\boldsymbol{\phi}\left(\mathbf{W}^{(l)}\mathbf{h}_{\mathrm{dep}_j}^{(l)}\left(\mathbf{h}_{\mathrm{sem}_i}^{(l-1)}\right)^{\mathrm{T}}\right)\right)}{\sum\limits_{k\in V_i}\exp\left(\boldsymbol{\phi}\left(\mathbf{W}^{(l)}\mathbf{h}_{\mathrm{dep}_k}^{l}\left(\mathbf{h}_{\mathrm{sem}_i}^{(l-1)}\right)^{\mathrm{T}}\right)\right)}, \tag{6}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{D\times F}$, $\mathbf{h}_{\mathrm{dep}_j}^{(l)} \in \mathbb{R}^F$ is the latest embedding of node *j*'s dependency relation updated at the *l* layer of the dependency relation GAT, $\mathbf{h}_{\mathrm{sem}_i}^{(l-1)} \in \mathbb{R}^D$.

The corresponding token embedding at *l*-th layer output by the token dependency GAT is:

$$\mathbf{h}_{\mathrm{sem}_i}^{(l)} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in V_i} u_j \beta_{ij}^{m(l)} \mathbf{W}^{m(l)} \mathbf{h}_{\mathrm{sem}_j}^{(l-1)}\right). \tag{7}$$

Eventually, the two type embeddings of node *i* output by the last layer (*L* layer) of GATs are concatenated as the final representation.

$$\mathbf{h}_i^{(L)} = \mathbf{h}_{\mathrm{sem}_i}^{(L)} \mid\mid \mathbf{h}_{\mathrm{dep}_i}^{(L)} \tag{8}$$

Classification

After the dependency structure-enhanced graph attention network (DSE-GAT), the output embedding of sentence *s* is $\mathbf{H}^{(L)} = \{\mathbf{h}_1^{(L)}, \cdots, \mathbf{h}_i^{(L)}, \cdots, \mathbf{h}_n^{(L)}\}$. Finally, we pour the embedding of each node into a fully connected network, which is followed by a softmax function to compute the distribution over the event type tags. The standard cross-entropy loss with weight is used as our objective function to strengthen the influence of event type tags, in which the label weight be computed according to the method in (Wan et al. 2023a,d).

Experiments and Results

Data and Evaluation Metric

We conducted experiments on ACE2005[1] English and Chinese corpus. ACE2005 is recognized as a benchmark dataset for event detection and has been extensively used in prior research. The StanfordCoreNLP tool[2] is used to generate the token dependency tree. For the evaluation metric, we considered the average $F1$ obtained from validation as the evaluation metric for our experimental results.

No.	Models	$F1$
1	DMCNN (Chen et al. 2015)	69.1
2	JRNN (Nguyen, Cho, and Grishman 2016)	69.3
3	ANN-AugAtt (Liu et al. 2017)	71.7
4	dbRNN (Sha et al. 2018)	71.9
5	HBTNGMA (Chen et al. 2018)	73.3
6	GCN-ED (Nguyen and Grishman 2018)	73.1
7	JMEE (Liu, Luo, and Huang 2018)	73.7
8	MOGANED (Yan et al. 2019)	75.7
9	GateGCN (Lai, Nguyen, and Nguyen 2020)	77.6
10	EE-GCN (Cui et al. 2020)	77.6
11	SA-GRCN (Liu, Xu, and Liu 2021)	78.0
12	AGGED (Xie and Tu 2022)	79.9
13	DualGAT (Mi, Hu, and Li 2022)	82.7
14	EKD (Tong et al. 2020)	78.6
15	S^2-JDN (Li et al. 2021)	79.5
16	Saliency ED (Liu, Chen, and Xu 2022)	75.8
17	HPNet (Huang et al. 2020)	77.8
18	DNR (Liao et al. 2021)	81.8
19	RLIL (Li et al. 2022)	80.7
20	GPTEDOT (Veyseh et al. 2021)	79.2
21	Seq2Seq (Xie et al. 2021)	82.8
22	**DSE-ED (Ours)**	**86.3**

Table 1: Performance comparison on ACE2005-English (baseline results are sourced from relevant research papers).

Hyper-Parameter and Baselines

The batch size, learning late, dropout, and iteration are set as 1,1e-4, 0.2, and 40, respectively. The embedding dimension of dependency relation type is set as 20. The number of GAT layers and attention heads are 2 and 3. The hidden layer size and layers of Bi-LSTM are 100 and 2.

To comprehensively evaluate our proposed DSE-ED, we compared it against a range of baselines and state-of-the-art models. Our comparisons for the ACE2005 English dataset contain models across three categories: sequence-based, GCN-based, and other neural network approaches (Table 1). Furthermore, recognizing that several of these models have also been evaluated using the ACE2005 Chinese corpus, we included them, along with additional models, as the baselines on the Chinese dataset (Table 2).

Overall Performance

We report our experimental results on the ACE2005 English and Chinese datasets in Tables 1 and 2. These results demonstrate that our DSE-ED achieves superior performance over all baseline models, with improvements

[1]https://catalog.ldc.upenn.edu/LDC2006T06

[2]https://stanfordnlp.github.io/CoreNLP/

No.	Models	$F1$
1	DMCNN(Char) (Chen et al. 2015)	57.8^*
2	DMCNN(Word) (Chen et al. 2015)	60.2^*
3	HBTNGMA(Char) (Chen et al. 2018)	45.5^*
4	HBTNGMA(Word) (Chen et al. 2018)	53.5^*
5	HNN (Feng, Qin, and Liu 2018)	63.0^*
6	NPN (Lin et al. 2018)	64.2^*
7	TLNN (Ding et al. 2019)	67.8^*
8	CAEE (Wu et al. 2022)	$73.1^\dagger$
9	EDM (Qin et al. 2022)	$76.2^\dagger$
10	**DSE-ED (Ours)**	**79.6**

Table 2: Performance comparison on ACE2005-Chinese. $*$ indicates results sourced from Ding et al. (2019); † indicates results sourced from corresponding papers.

of 3.5∼18.6 percentage points for the ACE2005 English dataset, and 3.4∼34.1 percentage points for the Chinese dataset. Moreover, we provide analyses of Table 1, focusing on the influence of various factors on performance, including (1) the token dependency structure, (2) the dependency relation, and (3) external knowledge and various strategies.

Token dependency structure. The comparison between sequence-based (lines 1–5) and GCN-based methods (lines 6–13) shows that the latter consistently outperforms the former. This trend suggests that token dependency structures are rich in informative content, and effectively leveraging these structures can enhance event detection performance. Specifically, MOGANED and GateGCN baselines attain this target by splitting the token dependency structure and filtering noise data in the token dependency tree.

Dependency relation. GCN-based approaches, which integrate dependency relation information (e.g., EE-GCN and SA-GRCN), have demonstrated superior performance over sequence-based models. SA-GRCN, in particular, enhances its effectiveness by dynamically adjusting the token dependency structure based on EE-GCN. In this paper, our DSE-ED not only adopts the token dependency structure, but also constructs the dependency relation graph to capture the structural information among dependency relations, resulting in $F1$=86.3% on ACE2005 English corpus.

External knowledge and various strategies. Recent neural network methods (lines 14–21) have further promoted the development of event detection by using external knowledge and opening up new detection strategies. Models such as EKD, S^2-JDK, and Saliency ED are based on the open-domain trigger knowledge, word-event co-occurrence frequencies, and trigger saliency attribution, respectively. Also, novel approaches like hierarchical policy networks, contrastive learning strategies, and reinforcement learning are explored in HPNet, DNR, and RLIL. Moreover, GPTE-DOT implemented event detection based on a pre-training model, while Seq2Seq converted event detection into graph parsing. Despite these diverse methodologies, a common limitation among these models is their underutilization of token dependency information, resulting in a worse performance than our DSE-ED.

No.	Graph Ablation	$F1$
1	DSE-ED	86.3
2	w/o TDG	74.1
3	w/o DRG	77.5
4	w/o TDG & DRG	71.2

Table 3: $F1$ score with graph ablations.

No.	Attention Strategies	$F1$
1	DSE-ED	86.3
2	DSE-ED (Non-Interaction)	83.5
3	DSE-ED (Sem)	82.6
4	DSE-ED (Dep)	83.0
5	DSE-ED (Original)	81.3

Table 4: $F1$ score under diverse attention strategies

Additional Analysis and Discussions

To investigate the impact of each component on event detection performance, we conducted additional experiments on the ACE2005 English corpus.

Ablation Study

To assess the impact of TDG and DRG within the DSE-ED framework, we conducted ablation experiments. The results, as reported in Table 3, indicate significant contributions of both graphs to the model's performance. Removing TDG (line 2) and DRG (line 3) led to 12.2 and 8.8 percentage points reduction, respectively. This suggests that the TDG has a more important influence on event detection. This is because token embeddings are taken from the pre-training language model and have rich semantic meanings, while the embeddings of nodes in DRG are generated by random initialization. The semantics needs to be gradually learned in the model training. When both TDG and DRG are excluded (line 4), the model achieves $F1$ of 71.2%), superior to some baselines in Table 1. The effectiveness can be attributed to DSE-ED's encoding of the dependency relation information and obtaining the sequential structure by Bi-LSTM.

Attention Coefficient

Now we explore the impact of different attention strategies in DSE-ED. In addition to the attention coefficient strategy proposed in this paper, we implemented four baseline strategies, and Table 4 presents their $F1$. In *DSE-ED (Non-Interaction)*, although the proposed attentions are used, the embeddings in the two graphs do not interact. That is, the output embeddings of the Bi-LSTM layer are used for the two GATs, instead of the latest output embeddings of GATs. *DSE-ED (Sem)* employs the original GAT attention mechanism in the TDG, while utilizing our proposed attention strategy in the DRG. Conversely, *DSE-ED (Dep)* mirrors this approach but applies the original GAT attention in the DRG and our attention strategy in the TDG. *DSE-ED (Original)* uses the original GAT attention in both graphs.

The comparison between lines 1 and 2 in Table 4 reveals a 2.8 percentage point decrease in performance when static

No.	Adjustments	$F1$
1	DSE-ED (All)	86.3
2	DSE-ED (Sub)	85.2
3	DSE-ED (Obj)	82.1
4	DSE-ED (Parallel)	83.2
5	DSE-ED (No Adjustment)	78.2

Table 5: $F1$ score with diverse core argument adjustments.

No.	Levels	$F1$
1	DSE-ED (TDG + DRG)	86.3
2	DSE-ED (TDG)	85.2
3	DSE-ED (DRG)	85.5
4	DSE-ED (No Level)	82.2

Table 6: $F1$ score when using the level in different graphs.

Bi-LSTM embeddings are used, showing the value of our interaction strategy. Lines 3 and 4 demonstrate the effects of unidirectional interaction within the graphs, each testing a different attention coefficient calculation approach. The results indicate that both proposed attention mechanisms are beneficial, with dependency relations playing a greater role in the interaction process than tokens. This further highlights the advantage of DRG, i.e., a better edge representation can be obtained by aggregating neighbors' features with a graph neural network.

Core Argument Adjustment

We investigated the effects of different core argument adjustments, as reported in Table 5. The variations include adjustments exclusively to the subject ('Sub'), object ('Obj'), and parallel clauses ('Parallel'), along with a scenario with no level adjustment ('No Adjustment'). According to Table 5, adjusting the level of core arguments significantly enhances event detection, outperforming other schemes by 1.1 to 8.1 percentage points in $F1$ score. Comparing these results, the hierarchy of adjustment importance appears to be the subject, followed by parallel clauses, and then the object. It confirms that core arguments in the subject position are most important for event detection, and elevating arguments in parallel clauses also contributes positively to the process.

The Level

Table 6 presents the results of incorporating hierarchical levels within the graph nodes of our DSE-ED model. Specifically, *DSE-ED (TDG)* and *DSE-ED (DRG)* denote the use of level features exclusively in the nodes of the TDG and the DRG, respectively. When levels are introduced in just one of the graphs (as observed in lines 2 and 3), there is a decrease in the recognition effect by 1.1 and 0.8 percentage points, respectively. These similar magnitudes of impact suggest that the level feature plays a comparable role in enhancing event detection across different graph structures. Comparison between lines 1 and 4 in Table 6 demonstrates the important role of hierarchical levels, showing a significant improvement in detection effectiveness when levels are applied.

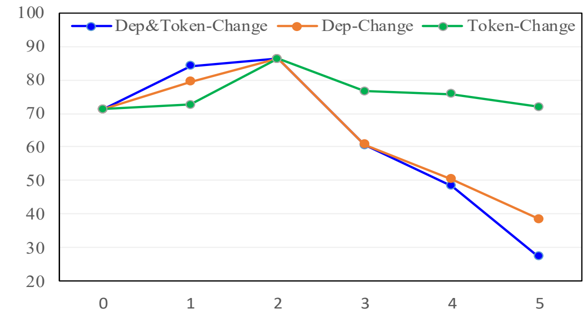


Figure 5: $F1$ score of DSE-ED with different numbers of GAT layers. The horizontal axis is the number of GAT layers. *Dep&Token-Change* indicates a scenario where both the token dependency graph attention network and the dependency relation graph attention network alter their layer counts simultaneously. *Dep-Change* presents the situation where only the dependency relation graph attention network's layer count varies, while the other network maintains its optimal layer count, which is 2 in this case. Similarly, *Token-Change* refers to altering only the token dependency graph attention network's layers, keeping the dependency relation graph at its optimal layer configuration.

GAT Layers

We examined how varying the number of GAT layers affects the performance of the DSE-ED in event detection. In Figure 5, we analyzed the $F1$ scores for 1 to 5 layers. It can be seen that the overall trend is an initial increase in performance followed by a decline, aligning with findings from previous studies (Wan et al. 2023a; Yang et al. 2019). The layers within the dependency relation graph attention network (Dep-Change) significantly influence event detection performance. The DRG's ability to reduce the feature aggregation distance between nodes contributes to this effect. However, as the number of layers increases, the over-smoothing issue becomes more pronounced, leading to a worse performance when compared to Token-Change. This observation shows the advantage of implementing DRG.

Conclusions

This paper presents a Dependency Structure-Enhanced Event Detection (DSE-ED) model, leveraging token dependency and dependency relation structures. We construct a Token Dependency Graph (TDG) using a syntactic parser and transform its dependency edges into nodes to create a Dependency Relation Graph (DRG). Then, the levels of core argument and its dependency relation nodes are raised to strengthen their roles. Subsequently, the combination of token and dependency relation embedding jointly regulates the attention coefficient in the graph attention network (GAT). Our experiments on the ACE2005 English and Chinese corpus demonstrate the model's effectiveness in enhancing event detection capability.

Acknowledgements

This research was partially supported by the National Natural Science Foundation of China (62272205, 61972184, 62272206, and 62076112), the Science & Technology Project of the Department of Education of Jiangxi Province (GJJ210531), the Natural Science Foundation of Jiangxi Province (20232ACB202008 and 20212ACB202002), and the Funding Program for Academic and Technical Leaders in Major Disciplines of Jiangxi Province (20213BCJL22041).

References

Ahmad, W. U.; Peng, N.; and Chang, K.-W. 2021. GATE: Graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 1207–1216.

Chen, Y.; Xu, L.; Liu, K.; Zeng, D.; and Zhao, J. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 167–176.

Chen, Y.; Yang, H.; Liu, K.; Zhao, J.; and Jia, Y. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1267–1276.

Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; and Shi, J. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*, 2329–2339.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186.

Ding, N.; Li, Z.; Liu, Z.; Zheng, H.; and Lin, Z. 2019. Event detection with trigger-aware lattice neural network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 347–356.

Feng, X.; Qin, B.; and Liu, T. 2018. A language-independent neural network for event detection. *Science China Information Sciences*, 61(9): 1–12.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Huang, P.; Zhao, X.; Takanobu, R.; Tan, Z.; and Xiao, W. 2020. Joint event extraction with hierarchical policy network. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2653–2664.

Lai, V. D.; Nguyen, T. N.; and Nguyen, T. H. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5405–5411.

Li, Q.; Peng, H.; Li, J.; Wu, J.; Ning, Y.; Wang, L.; Philip, S. Y.; and Wang, Z. 2022. Reinforcement learning-based dialogue guided event extraction to exploit argument relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 30: 520–533.

Li, R.; Zhao, W.; Yang, C.; and Su, S. 2021. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2625–2635.

Liao, J.; Zhao, X.; Li, X.; Zhang, L.; and Tang, J. 2021. Learning discriminative neural representations for event detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 644–653.

Lin, H.; Lu, Y.; Han, X.; and Sun, L. 2018. Nugget proposal networks for Chinese event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1565–1574.

Liu, A.; Xu, N.; and Liu, H. 2021. Self-attention graph residual convolutional networks for event detection with dependency relations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 302–311.

Liu, J.; Chen, Y.; and Xu, J. 2022. Saliency as evidence: Event detection with trigger saliency attribution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 4573–4585.

Liu, S.; Chen, Y.; Liu, K.; and Zhao, J. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1789–1798.

Liu, X.; Luo, Z.; and Huang, H. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1247–1256.

Mi, J.; Hu, P.; and Li, P. 2022. Event detection with dual relational graph attention networks. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, 1979–1989.

Nguyen, T. H.; Cho, K.; and Grishman, R. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 300–309.

Nguyen, T. H.; and Grishman, R. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 5900–5907.

Pouran Ben Veyseh, A.; Ngo Nguyen, T.; and Nguyen, T. H. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3651–3661.

Qin, Y.; Wang, Z.; Zhang, Y.; Chen, K.; and Zhang, M. 2022. Advancing Chinese event detection via revisiting character information. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(4): 1–9.

Sha, L.; Qian, F.; Chang, B.; and Sui, Z. 2018. 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)*, 5916–5923.

Tong, M.; Xu, B.; Wang, S.; Cao, Y.; Hou, L.; Li, J.; and Xie, J. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 5887–5897.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 1–12.

Veyseh, A. P. B.; Lai, V.; Dernoncourt, F.; and Nguyen, T. H. 2021. Unleash GPT-2 power for event detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 6271–6282.

Wan, Q.; Wan, C.; Hu, R.; and Liu, D. 2021. Chinese financial event extraction base on syntactic and semantic dependency parsing. *Chinese Journal of Computer*, 44(3): 508–530.

Wan, Q.; Wan, C.; Xiao, K.; Hu, R.; and Liu, D. 2023a. A Multi-Channel hierarchical graph attention network for open event extraction. *ACM Transactions on Information Systems (TOIS)*, 41(1): 1–27.

Wan, Q.; Wan, C.; Xiao, K.; Hu, R.; Liu, D.; and Liu, X. 2023b. CFERE: Multi-type Chinese financial event relation extraction. *Information Sciences*, 630: 119–134.

Wan, Q.; Wan, C.; Xiao, K.; Liu, D.; Li, C.; Zheng, B.; Liu, X.; and Hu, R. 2023c. Joint document-level event extraction via token-token bidirectional event completed graph. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL)*, 10481–10492.

Wan, Q.; Wan, C.; Xiao, K.; Xiong, H.; Liu, D.; and Liu, X. 2023d. Token-Event-Role structure-based multi-channel document-level event extraction. *arXiv preprint arXiv:2306.17733*.

Wu, X.; Wang, T.; Fan, Y.; and Yu, F. 2022. Chinese event extraction via graph attention network. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(4): 1–12.

Xie, J.; Sun, H.; Zhou, J.; Qu, W.; and Dai, X. 2021. Event detection as graph parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1630–1640.

Xie, Z.; and Tu, Y. 2022. A graph convolutional network with adaptive graph generation and channel selection for event detection. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, 11522–11529.

Yan, H.; Jin, X.; Meng, X.; Guo, J.; and Cheng, X. 2019. Event detection with multi-order graph convolution and aggregated attention. 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)*, 5765–5769.

Yang, Y.; Wang, X.; Song, M.; Yuan, J.; and Tao, D. 2019. SPAGAN: Shortest path graph attention network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 4099–4105.