2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191893

Semantic-Reconstructed Graph Transformer Network for Event Detection

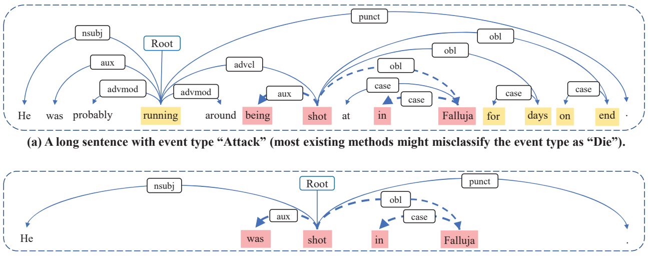
Zhuochun Miao, Xingrui Zhuo, Gongqing Wu* and Chenyang Bu*
Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China),
School of Computer Science and Information Engineering, Hefei University of Technology, Anhui, China

***Abstract*—Event detection (ED) is a key subtask of information extraction to extract key events, such as stock rise and fall and social public opinion, from news or social media. Although current GCN-based event detection methods achieve remarkable success via building graphs with dependency trees, they typically suffer from two challenges: 1) They use sequence models to learn contextual information of sentences, ignoring the long-term dependencies problem of sequence models might learn ineffective information and make it propagate in GCN layers. 2) Most methods do not exploit global dependency label information and grammatical structure information that convey rich linguistic knowledge directly, and only consider local dependency label information. To cope with these challenges, we propose a novel event detection model via semantic-reconstructed graph transformer networks (SRGTNED), which incorporates semantic reconstruction and path information collection methods. Using the semantic reconstruction method, we assign a pruned sequence to each word based on the path information to capture contextual information consistent with sentence semantics. Moreover, to better utilize global dependency label information and grammatical structure information, a Graph Transformer Network (GTN)-based heterogeneous graph embedding framework is introduced to automatically learn path information between important words by converting sentences as heterogeneous graphs. We conduct experiments on the ACE2005 dataset and the Commodity News dataset, and the experimental results demonstrate that our method significantly outperforms 11 state-of-the-art baselines in terms of the F1-score.**

***Index Terms*—event detection, semantic reconstruction, path information, heterogeneous graph.**

I. Introduction

Event detection is a subtask of information extraction that aims to identify specific types of events from given text. Generally, each event is labeled by a word or phrase called "event trigger" that clearly expresses the occurrence of events. The task of event detection is to find trigger words and correctly classify them as specific types. For sentence-level event detection, it is difficult to deal with the problem that the same trigger word triggers different event types. Taking Fig. 1 as an example, the event detection task should correctly detect the trigger word "shot" in two sentences and classify it to the event type "Attack" and "Die", respectively.


nsubj
aux
advmod
Root
advmod
advcl
aux
punct
obl
obl
obl
case
case
case
case
He was probably running around being shot at in Falluja for days on end .

(a) A long sentence with event type "Attack" (most existing methods might misclassify the event type as "Die").

(b) A short sentence with event type "Die", whose structure is similar to the local structure in (a).

Fig. 1. Example of two structurally similar sentences with different event types (dotted arrows and red boxes in (a) and (b) indicate the similar sentence structure). An edge (e.g., nsubj) connecting two words represents additional grammatical structure information for event detection. Because (a) has a local structure similar to (b), the existing methods tend to misclassify the type of (a) as "Die" when exploring the local context. In fact, a model should focus on the long-range words with yellow boxes in (a) for correctly classifying the type of (a) as "Attack".

Recently, because graph convolutional network (GCN) is capable of capturing long-range dependency information, it has been applied to the event detection task and achieved state-of-the-art performance. Existing GCN-based methods [1], [2] usually use the token embedding learned by a sequence model such as BiLSTM as GCN input, ignoring the effectiveness of its learned contextual information. As a result, the learned ineffective information might propagate in GCN layers. For example, in Fig. 1(a), the distance between "being" and event location "Falluja" is closer in the sentence. In the processing of learning the contextual information by a sequence model, the information of "being" and "Falluja" will be more propagated to the trigger word "shot". Under this learning process, the meaning of this sentence might be misjudged as "being shot at in Falluja" which is structurally similar in semantic to the Fig. 1(b). Specifically, the meaning of "shot" is misjudged as "death caused by shot" and the event type is misclassified as "Die". In fact, the event-related subject "He", action "running" and time "for days on end" which are far from the trigger word "shot" convey important information for understanding the semantic of the sentence correctly. In this example, the important information is lost during the information propagation process of the sequence model. Therefore, the correct meaning of "He was running being shot at for days on end" is lost, resulting in the event type not being correctly classified as "Attack".

Besides, most previous GCN-based methods only consider the dependency relations and ignore the dependency label information of edges in dependency trees [3], [4]. Although some recent studies [5], [6] take the dependency label information into consideration, they only consider dependency label information between connected nodes. These methods convert sentences into **homogeneous graphs** and obtain edge features according to an embedding table, as shown in Fig. 2(a). In these methods, edge information is used to only

*Corresponding authors

update the information of its connected nodes, not for updating global edge information and node information. In fact, the distance between the trigger word and other important words in the dependency tree is probably multi-hop. For example, the trigger word "shot" in the Fig.1 (a) is connected with "He" through the edge "advcl", the node "running" and the edge "nusbj". Here, "advcl" and "nusbj" represent the adverbial clause modifier and subject respectively. Obviously, the path composed of this "subject-predicate-adverbial" grammatical structure provides more information than a single edge with its two connected words, and the information contained in this path has a crucial impact on understanding the Fig.1 (a). Existing methods only use the local dependency label information between connected nodes, which might not be able to capture such rich path information for the example in Fig. 1(a). Therefore, how to learn the path information is crucial for understanding sentences.

To tackle these problems, this paper proposes Semantic-Reconstructed Graph Transformer Network for Event Detection, called SRGTNED. It utilizes semantic information and global dependency label information to enhance event detection. For the problem that ineffective contextual information learned by a sequence model will be wrongly propagated in GCN layers. Our method sorts words in a sentence according to the importance of contextual information of words to reconstruct this sentence. In this way, the contextual information of a sentence is fully utilized to alleviate the negative impact of the ineffective contextual information on our model. Specially, for the problem that the global dependency label information might not be utilized. We consider different dependency labels as different types of edges, and convert sentences into **heterogeneous graphs** (e.g., as shown in Fig. 1(a), "nsubj" and "advcl" represent different types of edges in a graph). Then, we design a Graph Transformer Network (GTN)-based [7] heterogeneous graph embedding framework to capture path information from heterogeneous graphs and fuse it into a feature space.

To summarize, as shown in Fig. 2(b), we propose SRGTNED which utilizes both path information collection and semantic reconstruction methods to improve event detection performance. Our contributions are summarized as follows:

- To alleviate the influence of ineffective contextual information, we propose a semantic reconstruction method based on token importance ranking to learn the effective contextual information of sentences from multiple perspectives.
- Unlike traditional methods of assigning truth values to each dependency label according to an embedding table, we design a GTN-based heterogeneous graph embedding framework for mining the grammatical features of different types of edges in dependency trees to learn the path information.
- Experiments conducted on the ACE2005 dataset and the Commodity News dataset show that SRGTNED significantly outperforms 11 state-of-the-art baselines in terms of F1-score.

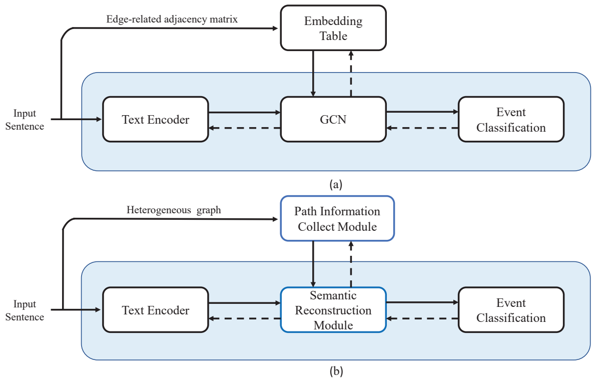


Fig. 2. Differences between traditional methods and our method in the way of obtaining node contextual information and edge information. (a) and (b) are the modular structures of traditional GCN-based ED methods and our proposed SRGTNED method, respectively. The solid arrows and the dashed arrows denote the forward pass of information and backpropagation of gradients, respectively.

II. Related Work

In earlier studies, researchers usually used pattern matching-based methods [8], [9], [10] and statistic feature engineering-based methods [11], [12], [13] to perform event detection tasks. Pattern matching-based methods aim to detect and extract information of events under the guidance of the event templates. Statistic feature engineering-based methods model statistical distributions based on statistical features to derive clustering of events. Although these methods achieve good performance in specific fields, it is not portable and poorly robust. With the development of deep learning, most researchers have proposed event detection methods based on sequence models [14], [15], [16], adversarial training [17] and knowledge base [18], [19].

In recent studies, researchers apply GCN to event detection and achieve state-of-the-art performance. For event detection, the graphs in GCN are built from dependency trees, thus dependency relations play an essential role in ED. [1] is the first to apply GCN on dependency trees, it combines syntatic dependencies between words for better aggregation of contextual information. [2] applies graph attention network on high-order graphs to compute high-order representions in order to capture more information and overcome the information vanishing problem caused by stacking multiple layers of GCN. [4] proposed a gating mechanism in ED to filter out noisy information to capture effective information. Later, researchers found that dependency labels provide rich syntactic information that could improve the performance of ED. [5] considers the importance of dependency labels and adds them to the information aggregation process between connected nodes. [6] designs an attention tensor to explore node-to-node dependency relations to aggregate effective information and introduces residual networks to solve the information vashing problem. [20] uses the shortest path method to obtain a subtree with contextual information, and performs node feature aggregation based on the shortest path. Although these methods try to aggregate effective contextual information, they

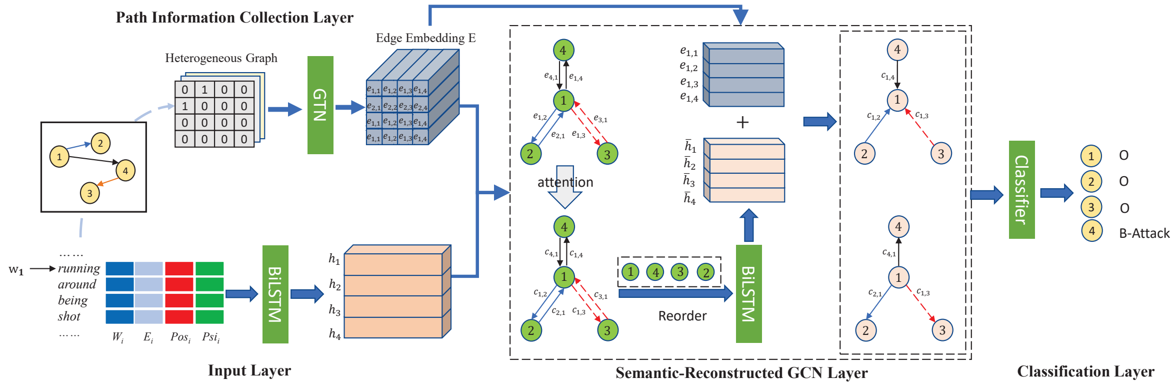


Fig. 3. Framework of Semantic-Reconstructed Graph Transformer Networks for Event Detection. There are two main modules of SRGTNED: The Path Information Collection Module obtain global dependency label information and grammatical information of each edge through specific path, and the Semantic Reconstruction Module obtain effective contextual information through node and path information.

all apply the solution to the information aggregation process of GCN, rather than capture effective contextual information before GCN to obtain good performance. Furthermore, [5] and [6] only use the local dependency label information without considering the path information, and the dependency label information is only used as the weight of the node aggregation process instead of being directly integrated into the feature space. How to capture effective contextual information before GCN layers and make full use of the global dependency label information is still a challenge in the event detection task.

III. Methods

The overall SRGTNED framework is shown in Fig. 3, which is mainly composed of four components: the Input Layer, the Path Information Collection Layer, the Senmantic-Reconstructed GCN Layer based on importance ranking and the Classification Layer.

A. Input Layer

Let $W = \{w_1, w_2, ..., w_n\}$ denote a sentence containing n words. We transform each w_i by concatenating word embedding $\mathbf{w}_i$, entity embedding $\mathbf{e}_i$, POS embedding $\mathbf{pos}_i$ and position embedding $\mathbf{ps}_i$ into a comprehensive embedding vector $\mathbf{x}_i = [\mathbf{w}_i, \mathbf{e}_i, \mathbf{pos}_i, \mathbf{ps}_i]$, $\mathbf{x}_i \in \mathbb{R}^{d_w+d_e+d_{pos}+d_{ps}}$, where d_w, d_e, d_{pos} and d_{ps} represent the dimensions of word embedding, entity embedding, part-of-speech embedding and position embedding, respectively. Then BiLSTM is utilized to capture the initial contextual information of each word. For simplicity, we denote the contextualized word representations as $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n]$, where $\mathbf{H} \in \mathbb{R}^{n \times d}$ is used as the initial node feature in SRGTNED.

B. Path Information Collection Layer

Making full use of edge information composed of global dependency labels is the key to improve the performance of ED further. The homogeneous graph-based EE-GCN constructs a type-related tensor to represent features of the edges by assigning real values according to an embedding table. SA-GRCN further expresses the type-related tensor as two asymmetric tensors on the basis of EE-GCN and designs an attention tensor to describe the node-dependency features. Different from these methods, we think it is important to make full use of global dependency label information to obtain path information from structures such as “subject-verb-object”, and “subject-copula-predicate” that can’t be obtained by a single edge. Thus, we need to take a heterogeneous graph approach to this path-related feature extraction task involving multiple types of edges. In this section, we first introduce how to convert sentences into heterogeneous graphs. Then we use a GTN-based heterogeneous graph embedding framework to automatically explore path features that have important grammatical information with these heterogeneous graphs.

Consider a heterogeneous graph with a total of L types of edges, and each edge of the heterogeneous graph is one of the L types. A heterogeneous graph can be represented by a set $\{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_L\}$ containing L adjacency matrices, each adjacency matrix corresponding to a different edge type. If there is an edge of type l from node i to node j, then $\mathbf{A}_l(i, j) = \mathbf{A}_l(j, i) = 1$. Specially, $\mathbf{A} = \sum_{l=1}^{L} \mathbf{A}_l$ is an adjacency matrix of a homogeneous graph. We first obtain a one-length-path feature $\mathbf{Q}_l \in \mathbb{R}^{n \times n \times c}$ through convolution, and $\mathbf{Q}_l$ has weighted edges between any two nodes with edges in the L original matrices:

$$\mathbf{Q}_l = \sum_{t_l=1}^{L} \alpha_{t_l}^{(l)} \mathbf{A}_{t_l}, \tag{1}$$

where $\alpha_{t_l}^{(l)}$ is the weight of the l-th edge type t_l in the l-th layer, and c is the number of channels. We obtain the dependency label information between connected nodes by this transformer mechanism. We further multiply $\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_P$ in order to get the multi-hop path information $\mathbf{E} \in \mathbb{R}^{n \times n \times c}$ composed of global dependency label information and grammatical structure information.

$$\begin{aligned} E &= \mathbf{Q}_1 * \mathbf{Q}_2 * ... * \mathbf{Q}_P \\ &= \sum_{t_1=1}^{L} \alpha_{t_1}^{(l)} \mathbf{A}_{t_l} \sum_{t_2=1}^{L} \alpha_{t_2}^{(l)} \mathbf{A}_{t_l} ... \sum_{t_1=1}^{L} \alpha_{t_P}^{(l)} \mathbf{A}_{t_P}. \end{aligned} \tag{2}$$

In this way, E contains path representations for identifying useful grammatical structures and dependency label information for multi-hop connections.

C. Semantic-Reconstructed GCN Layer

In this subsection, we first introduce the vanilla GCN model and then present the proposed Semantic-Reconstructed (SR) GCN.

GCN designs a method of extracting features from graph data, which has been widely used in graph node classification [21], [22], graph classification [23], [24], link prediction [25], [26], and graph representation learning [27], [28]. For event detection, a graph in vanilla GCN is built from a dependency tree and is represented by an adjacency matrix $\mathbf{A} = (a_{i,j})$. If there is a dependency edge between node i and node j, we assign $a_{i,j} = 1$; otherwise, $a_{i,j} = 0$. Based on $\mathbf{A}$, at the l-th layer of GCN, if we represent $\mathbf{H}^{(l-1)}$ as the input state, and $\mathbf{H}^{(l)}$ as the output state, the graph convolutional operation can be formulated as:

$$\mathbf{H}^{(l)} = \sigma(\mathbf{A}\mathbf{H}^{(l-1)}\mathbf{W}), \tag{3}$$

where $\mathbf{W}$ is a learnable convolutional filter and σ donates the ReLU activation function.

In vanilla GCN-based methods for event detection, the contextual information obtained through a sequence model is usually input into GCN. However, vanilla GCN models can't automatically aggregate the effective contextual information and the propagation of ineffective contextual information in vanilla GCN models will degrade event detection performance. Moreover, vanilla GCN-based models consider connected nodes equally and are not able to distinguish the importance of different dependency relations; thus pruning nodes and edges in graphs is important for ED.

To solve these problems, we propose SR-GCN. It firstly utilizes both node features and path features to calculate the importance scores of different dependency relations, then reorders the sequence based on these scores to learn effective information from multiple perspectives, finally fuse multiple ordered sequence information learned in LSTM and input it to GCN to obtain fianl representation. In this way, the contextual information of sentences is fully utilized, and the negative impact of the contextual information learned from sequence models on our model is alleviated. Specifically, we firstly utilize path features and node features to calculate the importance scores between nodes to prune graphs [29]. At the l-th layer of GCN, the importance scores $c_{i,j}^{(l)}$ between word w_i and w_j is calculated as:

$$c_{i,j}^{(l)} = \frac{a_{i,j} \cdot \exp(\mathbf{path}_i^{(l)} \cdot \mathbf{path}_j^{(l)})}{\sum_{j=1}^{n} a_{i,j} \cdot \exp(\mathbf{path}_i^{(l)} \cdot \mathbf{path}_j^{(l)})}, \tag{4}$$

where $a_{i,j} \in \mathbf{A}$, "$\cdot$" denotes the vector dot product operation. $\mathbf{path}_i^{(l)}$ and $\mathbf{path}_j^{(l)}$ are path features of w_i and w_j in the l-th layer of GCN, respectively. Specifically, these features are calculated by Eqs. (5) and (6):

$$\mathbf{path}_i^{(l)} = [\mathbf{h}_i^{(l-1)} || \mathbf{e}_{i,j}], \tag{5}$$

$$\mathbf{path}_j^{(l)} = [\mathbf{h}_j^{(l-1)} || \mathbf{e}_{j,i}]. \tag{6}$$

Here, $\mathbf{e}_{i,j} \in \mathbf{E}$, "$||$" denotes the vector concatenation operation. In this way, we get the connected nodes that are important for each node, instead of binary choice for $a_{i,j}$, to make better use of them. After that, we sort the importance scores in descending order $order(i)$ centered on w_i:

$$order(i) = \{c_{i,(1)}^{(l)}, c_{i,(2)}^{(l)}, ..., c_{i,(n)}^{(l)}\}. \tag{7}$$

According to Eq. (7), we set $order(i)$ to be the semantic-reconstructed token sequence with w_i as the center word, and the sequence of features [30] can be formulated as:

$$\mathbf{H}_{order(i)}^{(l-1)} = [\mathbf{h}_{(1)}^{(l-1)}, \mathbf{h}_{(2)}^{(l-1)}, ..., \mathbf{h}_{(n)}^{(l-1)}]. \tag{8}$$

By sorting, we center on each node and place important connected nodes at the front of the sequence to alleviate information vanish caused by the problem of long term dependencies in the sequence model, and learn more useful information by using the interaction between important nodes. Therefore, we input $\mathbf{H}_{order(i)}^{(l-1)}$ into LSTM to get node features $\bar{\mathbf{H}}_{order(i)}^{(l-1)}$ that learn the effective information centered on w_i:

$$\bar{\mathbf{H}}_{order(i)}^{(l-1)} = LSTM(\mathbf{H}_{order(i)}^{(l-1)}). \tag{9}$$

Then, we apply the adjacency matrix $\mathbf{C}^{(l)}$ formed by the importance scores $c_{i,j}^{(l)}$, the initial residual and identity mapping operation [31] in the graph aggregation process to get the representation of all tokens under $order(i)$:

$$\mathbf{H}_{order(i)}^{(l)} = \sigma(((1-\alpha)\mathbf{C}^{(l)}\tilde{\mathbf{H}}_{order(i)}^{(l-1)} + \alpha\mathbf{H}^{(0)})((1-\beta_l)\mathbf{I}_n + \beta_l\mathbf{W}^{(l)})), \tag{10}$$

$$\tilde{\mathbf{H}}_{order(i)}^{(l-1)} = \bar{\mathbf{H}}_{order(i)}^{(l-1)} + \mathbf{W}_E^{(l)}\mathbf{E}, \tag{11}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d_1}$ is the trainable weight matrix of the l-th layer; α and β are hyperparameters. α can balance the input and output, and β can balance the information gain and loss; σ is the ReLU activation function; $\mathbf{W}_E^{(l)}$ maps the path information embedding $\mathbf{E}$ to the same dimension as $\bar{\mathbf{H}}_{order(i)}^{(l-1)}$. Finally, to fuse the features of each word in all orders, we put the output of all orders into the $pooling$ layer to get the final effective representation:

$$\mathbf{H}^{(l)} = \sum_{i=1}^{n} \mathbf{H}_{order(i)}^{(l)}. \tag{12}$$

D. Classification Layer

We consider event detection as a multi-classification task using BIO annotations. After aggregating the token representations from each layer of the Semantic-Reconstructed Layer, we finally feed each token's representation into a fully connected layer. Then we use a softmax function to calculate the probability distribution $\mathbf{y}_i$ of each token w_i on all the label types:

$$\mathbf{y}_i = softmax(\mathbf{W}_t\mathbf{h}_i + \mathbf{b}_t), \tag{13}$$

where $\mathbf{W}_t \in \mathbb{R}^{d \times 2N_e+1}$, N_e denotes the number of event types, and $\mathbf{b}_t$ is a bias term. After softmax, event label with the largest probability is chosen as the classification result.

TABLE I
PERFORMANCE ON THE ACE2005 DATASET

Model	Precision	Recall	F1-score
GCN-ED [1] (2018)	77.9	68.8	73.1
JMEE [3] (2018)	76.3	71.3	73.7
MOGANED [2] (2019)	**79.5**	72.3	75.7
DMBERT [35] (2019)	77.9	72.5	75.1
EE-GCN [5] (2020)	76.7	78.6	77.6
GatedGCN [4] (2020)	78.8	76.3	77.6
EKD [36] (2020)	79.1	78.0	78.6
SA-GRCN [6] (2021)	78.6	77.4	78.0
SRGTNED	79.3	**79.9**	**79.6**

E. Training

Since the number of "O" labels is much larger than the number of other event labels, we introduce focal loss to solve the severe class imbalance problem:

$$J(\theta) = -\sum_{i=1}^{N_s}\sum_{j=1}^{N_{(i,w)}} (1-p(y_i^t|s_i,\theta))^{\gamma} log(p(y_i^t|s_i,\theta)) \cdot I(O) \\ + \lambda(1-p(y_i^t|s_i,\theta))^{\gamma} log(p(y_i^t|s_i,\theta)) \cdot (1-I(O)), \tag{14}$$

where y_i^t denotes the probability of assigning label t to word w_i, N_s denotes the number of sentences, $N_{(i,w)}$ denotes the total number of words in the i-th sentence; λ is the bias weight, if the label of a word is "O", then $I(O) = 1$, otherwise, $I(O) = 0$. γ can control the problem of the unbalanced number between simple and indistinguishable samples.

IV. EXPERIMENTAL RESULTS

A. Dataset, Resources and Evaluation Metrics

We perform experiments on two English benchmark datasets, namely, the ACE2005 dataset and the Commodity News dataset. The ACE2005 dataset is a widely used dataset for ED, which contains 599 documents and includes 33 event types. In contrast, the Commodity News dataset is a newly published dataset, which contains 150 documents and includes 18 event types. The Stanford CoreNLP and the Brat annotation toolkit are used for dependency parsing of the ACE2005 dataset and the Commodity News dataset, respectively. We use the same data split as previous works on ED for the ACE2005 dataset [32], [33], [34] and the Commodity News dataset [20]. We evaluate our model using Precision, Recall, and F1-score.

B. Hyperparameter Setting

We turn the parameters of our model on the validation set. The pre-trained word embeddings are trained by using the Skip-Gram algorithm on the NYT corpus. The dimension of word embedding, entity embedding, POS-tag embedding and position embedding is 100, 50, 50, 50, respectively. The hidden size of BiLSTM is set to 125. We set the layer of SRGCN to 3 and the hidden size of SRGCN to 150. For the word embedding, the parameter p of dropout is set to 0.5. For the hidden states, the parameter p of dropout is set to 0.3. The layer of GTN is set to 2, and the number of channels is set to 10. We set $\alpha = 0.1$ and $\beta = 0.5$. The detailed analysis of α and β is given in the following section. During training, parameter optimization is performed using the Adam algorithm with an initial learning rate 0.001 and the batch size 32. We adopt the CosineAnnealingWarmRestarts algorithm to update the learning rate with the first restart iteration 30. We set the bias weight λ to 5. We utilize a fixed maximum sentence length $n = 50$ by padding short sentences and cutting longer ones. We ran all the experiments using Pytorch 1.10.0 on a workstation with an Nvidia GeForce RTX 3060 GPU and an Intel Core i9-10900F CPU.

TABLE II
PERFORMANCE ON THE COMMODITY NEWS DATASET

Model	Precision	Recall	F1-score
Bert+BiLSTM [37]	75.0	71.0	70.0
GCN-ED [1] (2018)	92.0	94.0	92.0
JMEE [3] (2018)	89.0	91.0	90.0
MOGANED [2] (2019)	93.5	94.8	94.2
GCN-with contextual sub-tree [20] (2021)	91.0	**97.0**	93.0
GCN-with LCA sub-tree [20] (2021)	93.0	95.0	94.0
SRGTNED	**93.8**	94.9	**94.4**

C. Baseline

The following state-of-the-art methods are compared with our method: Bert+BiLSTM [37], a simple deep learning method, which uses a BiLSTM model with Bert to improve the performance for ED. GCN-ED [1], which uses GCN on syntactic dependency trees for ED and proposes argument-aware pooling. JMEE [3], which enhances GCN with highway connection and self-attention mechanism to improve the perfomance for ED. MOGANED [2], which uses GAT on multi-order graph and ulitizes attention mechanism to capture the contextual information. DMBERT [35], which applies adversarial training and autiomatically label the data to improve the performance for ED. EE-GCN [5], which utilizes dependency label of each edge during graph convolution. GatedGCN [4], which proposes a gating mechanism and introduces novel mechanisms to achieve the contextual diversity for the gates and the importance score consistency for the graphs and models in ED. EKD [36], which introduces open-domain trigger knowledge to improve the perfomance of ED. SA-GRCN [6], which uses attention mechanism to fuse syntactic structures and potential dependencies and applies residual connection in ED. GCN with LCA sub-tree [20], which uses GCN with the shortest dependency path between trigger candidate and entity candidate. GCN with contextual sub-tree [20], which uses GCN with length-one dependency path between trigger candidate and entity candidate.

D. Overall Performance

Our experimental results on the ACE2005 dataset are reported in Table 1. It shows that SRGTNED achieves the best F1-score and Recall with 1.0% and 1.3% improvement, respectively. Moreover, SRGTNED also maintains a close precision score compared to the state-of-the-art methods. We attribute the performance gain on two aspects: 1) The introduction of the path information collection method. Compared with the baselines EE-GCN and SA-GRCN which also introduce the edges features, our model still achieves a higher F1-score with 2.0% and 1.6% improvement, respectively. The reason for this result is that EE-GCN and SA-GRCN only utilize the edge embedding between the connected nodes, ignoring the rich information provided by the complete grammatical information in the sentence. In this way, these dependency label information collection modules work locally, not globally. By contrast, we utilize a GTN-based heterogeneous graph embedding framework to explore the path information composed of global dependency label information. Through the feature fusion of node information and path information, the information provided by the dependency labels can be used globally. Therefore, the path information collection method improves ED performance. 2) The introduction of the semantic reconstruction method. Compared with existing GCN-based methods, which input the contextual information learned by sequence models to GCN layers, our model achieves a higher F1-score with 1.0% improvement. The reason for this result is that these GCN-based methods ignore the effectiveness of the contextual infomation learned by sequence models. As a result, the ineffective contextual information is propagated in GCN layers. By contrast, we first rank the importance of nodes, and perform semantic reconstruction of sentences based on this ranking. Then, learning the contextual information of sentences from multiple perspectives to obtain correct information. In this way, the semantic reconstruction method improves the ED performance.

We also observe that SRGTNED gains improvement in Recall. We consider this is attributed to the joint action of our proposed path information collection method and semantic reconstruction method. The path information collection method provides global dependency label information to capture the relationship between nodes, and the semantic reconstruction method enables the model to understand sentence senmantics correctly. In this way, more triggers would be detected. Moreover, SRGTNED also maintains a close precision score compared to MOGANED. It is also attributed to our proposed two methods, which enable the model to classify trigger words correctly when detecting them.

To verify the generality of SRGTNED, we also report our experimental results on the Commodity News dataset in Table 2. We find that SRGTNED achieves the best F1-score and the best Precision by 0.2% and 0.3%, respectively. The reason is that the proposed semantic reconstruction method enables the model to correctly understand the semantics of sentences in different situations, so as to detect trigger words and classify them correctly.

E. Ablation Study

To demonstrate the effectiveness of the semantic reconstruction module and the path information collection module, we conduct ablation studies on the ACE2005 dataset as Table 3 shows:

TABLE III
Ablation Study on the ACE2005 dataset

Model	Precision	Recall	F1-score
-SR	78.6	78.2	78.4
-PIC	79.0	79.1	79.1
-PIC+Edge Embedding	76.7	79.4	78.0
-PIC&SR	78.7	77	77.8
Full SRGTNED	**79.3**	**79.9**	**79.6**

- -Semantic-Reconstructed (SR) Module: We remove the SR module and directly use the contextual information learned by BiLSTM as input to GCN. We find that the F1-score drops by 1.2%, which indicates that the SR module can learn effective contextual information and plays the most important role in our approach.
- -Path Information Collect (PIC) Module: We remove the PIC module and only use the node features to compute the attention scores. Moreover, the path information is also removed at the feature fusion stage before GCN. Lack of the path information results in a 0.5% drop of F1-score, demonstrating the effectiveness of the PIC module.
- -PIC+Edge Embedding: We remove the PIC module and assign real values to edges based on an embedding table. It is observed that F1-score drops by 1.6%, which demonstrates that the PIC module is more effective than Edge Embedding
- -SR&PIC: We remove both the SR module and the PIC module. In this scenario, the model gets the lowest performance with F1-score dropped by 1.8% in the ablation study, which suggests that both the PIC module and the SR module working together in our model can improve the performance.

Finally, in all these situations, four modified models get lower performance than SRGTNED, which suggests that both the path information collection method and semantic reconstruction method play important roles in our model.

F. Case Study

To give an intuitive understanding of our method, we perform case studies with our method and previous methods on the ACE2005 dataset, the results are shown in Table 4.

- For the first case, only MOGANED incorrectly identifies the word “telephone” as a trigger of the “Phone-write” event. We think it’s misled by the word “said” which is the center in the sentence. Lack of the edge information “nmod” which express how “said” happened from “said” to “telephone”, the “telephone” in the graph is considered as an unimportant node. Other methods use edge information which contains rich syntactic information to help judge the sentence semantics, so the prediction is correct.
- For the second case, EE-GCN correctly identifies the word “reaching” as a trigger, but incorrectly classifies it as an “Attack” event. It is because EE-GCN only utilizes the edge information between connected nodes, and misunderstands the meaning of the sentence as “reaching Is-

TABLE IV

CASE STUDIES OF PREVIOUS APPROACHES AND OUR APPROACH ON THE ACE2005 DATASET (THE TRIGGER WORD AND ERROR EVENT TYPES ARE IN BOLD BODY AND BLUE, RESPECTIVELY)

ID	CASES IN THE DATASET	MOGANED [2]	EE-GCN [5]	SRGTNED-SR	SRGTNED	Ground-Truth
1	The mob dragged out three members of a family and (killed) them with machetes and spears before fleeing the area, ẗhe spokesman said by **telephone** from Agartala.	None	Contact: Phone-Write	Contact: Phone-Write	Contact: Phone-Write	Contact: Phone-Write
2	Other units moved into airfield complexes in western Iraq believed to have Scud missiles capable of **reaching** Israel, and possibly weapons of mass destruction.	None	Movement: Transport	Conflict: Attack	Conflict: Attack	Conflict: Attack
3	British Chancellor of the Exchequer Gordon Brown on Tuesday **named** the current head of the country's energy regulator as the new chairman of finance watchdog the Financial Services Authority (FSA).	Personnel: Start-Position	Personnel: Start-Position	Personnel: Start-Position	Personnel: Nominate	Personnel: Nominate
4	Since then, many of Milosevic's political and business allies as well as his two children have been **accused** of crimes.	Justice: Charge-Indict	Justice: Charge-Indict	Justice: Charge-Indict	Justice: Charge-Indict	None

rael". Our method utilizes the path information, therefore our model with the PIC module understands the meaning of the sentence as "missiles reaching Israel" and correctly classifies it as an "Attack" event.

- For the third case, only our model correctly identifies the word "named" as a trigger of a "Nominate" event. This sentence can easily be misclassified because "named" is followed by position. The correct classification is attributed to the SR module that can learn contextual information from multiple perspectives, so as to understand the semantic of the sentence correctly.
- For the fourth case, all the methods identify the word "accused" as a trigger of a "Charge-Indict" event, but the ground truth is "None". It is possible that the sentence is not marked as a "Charge-Indict" event because the sentence contains only "Charge" and lacks the act of taking formal legal action as "Indict". We think this type of mistakes is reasonable and hard to deal with in the current ED framework.

G. *Parameter Sensitivity Analysis*

We apply the initial residual operation and the identity mapping operation with parameters α and β in the graph aggregation process. These two parameters α and β are ulitized to banlance the input and output and alleviate the information vanishing problem, respectively. In this section, we study how the value of α and β influence the event detection performance. We conduct experiments with α and β in the set $\{0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.3\}$ and the set $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, respectively. As Fig. 4 shows, we find that F1-score rises with increasing α at the beginning, and then decreases after reaching 0.1. In contrast, we find that F1-score decreases as β increases at the beginning, but suddenly reaches the highest value at $\beta = 0.5$, and then decreases continuously. We think this should be considered from two aspects. On the one hand, the updated node information probably has more information to correctly perform the event detection task than the initial information, so our model achieves a good effect when $\alpha = 0.1$. On the other hand, the information of the node itself has the same importance as the information of its connected nodes, so F1-score achieves the highest value when $\beta = 0.5$.

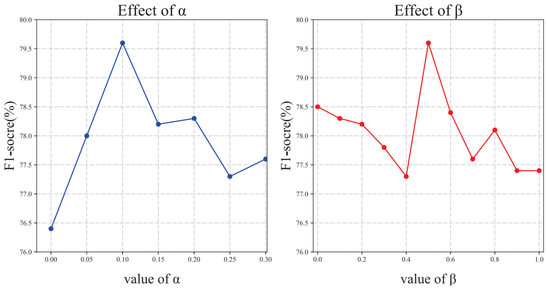


Fig. 4. F1-score variation with parameters α and β on the ACE2005 dataset

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel model named Semantic-Reconstructed Graph Transformer Network for Event Detection to alleviate the influence of ineffective contextual information in the aggregation process of GCN layers and fully utilize path information. Specifically, SRGTNED uses the semantic reconstruction method to learn effective contextual information and introduces the path information collection method to capture path information composed of global dependency label information on the basis of a heterogeneous graph embedding framework. We conduct experiments on the ACE2005 dataset and the Commodity News dataset, and the results well demonstrate our model's effectiveness.

In the future, we will further improve our model on the following aspects: 1) SRGTNED is currently only used for sentence-level event detection in English, we will try to improve it and use it in cross-language and cross-sentence scenarios. 2) Since there are few public datasets for event detection, we can try to introduce semi-supervised and unsupervised methods to alleviate the problem of consuming a lot of time to annotate datasets.

ACKOWNLEDGMENTS

Details omitted for double-blind reviewing. This work was supported by the National Natural Science Foundation of China (No. 62120106008), the Fundamental Research Funds for the Central Universities (No. JZ2022HGTB0239), and the Open Foundation Project of Anhui Province Key Laboratory of Infrared and Cryogenic Plasma (No. IRKL2022KF06).

REFERENCES

[1] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Proceedings of AAAI*. AAAI Press, 2018, pp. 5900–5907.

[2] H. Yan, X. Jin, X. Meng, J. Guo, and X. Cheng, "Event detection with multi-order graph convolution and aggregated attention," in *Proceedings of EMNLP-IJCNLP*. ACL, 2019, pp. 5765–5769.

[3] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proceedings of EMNLP*. ACL, 2018, pp. 1247–1256.

[4] V. D. Lai, T. N. Nguyen, and T. H. Nguyen, "Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks," in *Proceedings of EMNLP*. ACL, 2020, pp. 25 405–5411.

[5] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang, and J. Shi, "Edge-enhanced graph convolution networks for event detection with syntactic relation," in *Proceedings of EMNLP*. ACL, 2020, pp. 2329–2339.

[6] A. Liu, N. Xu, and H. Liu, "Self-attention graph residual convolutional networks for event detection with dependency relations," in *Proceedings of EMNLP*. ACL, 2021, pp. 302–311.

[7] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proceedings of NeurIPS*. ACL, 2019, pp. 11 960–11 970.

[8] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proceedings of ACL*. ACL, 2008, pp. 254–262.

[9] K. Cao, X. Li, M. Fan, and R. Grishman, "Improving event detection with active learning," in *Proceedings of RANLP*. ACL, 2015, pp. 72–77.

[10] K. Cao, X. Li, and R. Grishman, "Improving event detection with dependency regularization," in *Proceedings of RANLP*. ACL, 2015, pp. 78–83.

[11] S. Patwardhan and E. Riloff, "A unified model of phrasal and sentential evidence for information extraction," in *Proceedings of EMNLP*. ACL, 2009, pp. 151–160.

[12] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proceedings of ACL*. ACL, 2010, pp. 789–797.

[13] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu, "Using cross-entity inference to improve event extraction," in *Proceedings of ACL*. ACL, 2011, pp. 1127–1136.

[14] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of ACL-IJCNLP*. ACL, 2015, pp. 167–176.

[15] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *Proceedings of ACL-AFNLP*. ACL, 2015, pp. 365–371.

[16] L. Sha, F. Qian, B. Chang, and Z. Sui, "Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction," in *Proceedings of AAAI*. AAAI Press, 2018, pp. 5916–5923.

[17] Y. Hong, W. Zhou, J. Zhang, Q. Zhu, and G. Zhou, "Self-regulation: Employing a generative adversarial network to improve event detection," in *Proceedings of ACL*. ACL, 2018, pp. 515–526.

[18] S. Liu, Y. Chen, S. He, K. Liu, and J. Zhao, "Leveraging framenet to improve automatic event detection," in *Proceedings of ACL*. ACL, 2018, p. 2134–2143.

[19] Y. Chen, S. Liu, X. Zhang, K. Liu, and J. Zhao, "Automatically labeled data generation for large scale event extraction," in *Proceedings of ACL*. ACL, 2017, pp. 409–419.

[20] M. Lee, L.-K. Soon, and E.-G. Siew, "Effective use of graph convolution network and contextual sub-tree for commodity news event extraction," in *Proceedings of ECONLP*. ACL, 2021, pp. 69–81.

[21] W. Jin, T. Derr, Y. Wang, Y. Ma, Z. Liu, and J. Tang, "Node similarity preserving graph convolutional networks," in *Proceedings of WSDM*. ACM, 2021, pp. 148–156.

[22] K. Zhou, Q. Song, X. Huang, D. Zha, N. Zou, and X. Hu, "Multi-channel graph neural networks," in *Proceedings of IJCAI*. ijcai.org, 2020, pp. 1352–1358.

[23] K. Kong, G. Li, M. Ding, Z. Wu, C. Zhu, B. Ghanem, G. Taylor, and T. Goldstein, "Robust optimization as data augmentation for large-scale graphs," in *Proceedings of CVPR*. IEEE, 2022, pp. 60–69.

[24] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of CVPR*. IEEE, 2017, pp. 29–38.

[25] W. Lin, S. Ji, and B. Li, "Adversarial attacks on link prediction algorithms based on graph neural networks," in *Proceedings of AsiaCCS*. AAAI Press, 2020, pp. 370–380.

[26] X. Li, Y. Shang, Y. Cao, Y. Li, J. Tan, and Y. Liu, "Type-aware anchor link prediction across heterogeneous networks based on graph attention network," in *Proceedings of AAAI*. AAAI Press, 2020, pp. 147–155.

[27] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of CIKM*. ACM, 2015, pp. 891–900.

[28] S. Agarwal, K. Branson, and S. J. Belongie, "Higher order learning with graphs," in *Proceedings of ICML*. ACM, 2006, pp. 17–24.

[29] Y. Tian, G. Chen, Y. Song, and X. Wan, "Dependency-driven relation extraction with attentive graph convolutional networks," in *Proceedings of ACL-IJCNLP*. ACL, 2021, pp. 4458–4471.

[30] M. Chatzianastasis, J. F. Lutzeyer, G. Dasoulas, and M. Vazirgiannis, "Graph ordering attention networks," *arXiv e-prints*, pp. arXiv–2204, 2022.

[31] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of ICML*. PMLR, 2020, pp. 1725–1735.

[32] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of NAACL*. ACL, 2016, pp. 300–309.

[33] S. Liu, Y. Chen, K. Liu, and J. Zhao, "Exploiting argument information to improve event detection via supervised attention mechanisms," in *Proceedings of ACL*. ACL, 2017, pp. 1789–1798.

[34] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, "Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms," in *Proceedings of EMNLP*. ACL, 2018, pp. 1267–1276.

[35] X. Wang, X. Han, Z. Liu, M. Sun, and P. Li, "Adversarial training for weakly supervised event detection," in *Proceedings of NACCL*. ACL, 2019, pp. 998–1008.

[36] M. Tong, B. Xu, S. Wang, Y. Cao, L. Hou, J. Li, and J. Xie, "Improving event detection via open-domain trigger knowledge," in *Proceedings of ACL*. ACL, 2020, pp. 5887–5897.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NACCL*. ACL, 2019, pp. 4171–4186.