Cui SY, Yu BW, Cong X *et al*. Label-aware Chinese event detection with heterogeneous graph attention network. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(1): 227-242 Jan. 2024. DOI: 10.1007/s11390-023-1541-6

Label-Aware Chinese Event Detection with Heterogeneous Graph Attention Network

Shi-Yao Cui[1, 2] (崔诗尧), Bo-Wen Yu[1, 2] (郁博文), Xin Cong[1, 2] (从　鑫)
Ting-Wen Liu[1, 2, *] (柳厅文), *Member, CCF*, Qing-Feng Tan[3] (谭庆丰), *Member, CCF, IEEE*
and Jin-Qiao Shi[4] (时金桥)

[1] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100190, China*

[2] *School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China*

[3] *Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*

[4] *School of Cyber Security, Beijing University of Posts and Telecommunications, Beijing 100088, China*

E-mail: cuishiyao@iie.ac.cn; yubowen@iie.ac.cn; congxin@iie.ac.cn; liutingwen@iie.ac.cn; tqf528@gzhu.edu.cn
shijinqiao@bupt.edu.cn

Received April 23, 2021; accepted January 2, 2023.

Abstract　Event detection (ED) seeks to recognize event triggers and classify them into the predefined event types. Chinese ED is formulated as a character-level task owing to the uncertain word boundaries. Prior methods try to incorporate word-level information into characters to enhance their semantics. However, they experience two problems. First, they fail to incorporate word-level information into each character the word encompasses, causing the insufficient word-character interaction problem. Second, they struggle to distinguish events of similar types with limited annotated instances, which is called the event confusing problem. This paper proposes a novel model named Label-Aware Heterogeneous Graph Attention Network (L-HGAT) to address these two problems. Specifically, we first build a heterogeneous graph of two node types and three edge types to maximally preserve word-character interactions, and then deploy a heterogeneous graph attention network to enhance the semantic propagation between characters and words. Furthermore, we design a pushing-away game to enlarge the predicting gap between the ground-truth event type and its confusing counterpart for each character. Experimental results show that our L-HGAT model consistently achieves superior performance over prior competitive methods.

Keywords　Chinese event detection, heterogeneous graph attention network (HGAT), label embedding

1　Introduction

Event detection (ED) aims to locate event triggers from raw texts and classify them into the predefined event types. Generally, triggers are those words which evoke events of interest. For example, the word “visit” may be a strong signal of an event of the “Meet” type. ED is an important task for various downstream applications, such as document summarization[1], knowledge base population[2, 3], and question answering[4, 5].

Recent studies mainly focus on English ED and formulate it as a word-level task of multi-class classification paradigm[6–11] or sequence labeling paradigm[12–14]. However, Chinese ED is more tricky due to the absence of explicit word delimiters in Chinese texts. Directly casting Chinese ED as a character-level task ignores the meaningful semantics contained in lexcial words. For example, the instances of character “投” in the word “投资 (invest)” and word “投篮 (throw)” have entirely different meanings and trigger different types of events. Therefore, some methods

Regular Paper

This work was supported by the National Key Research and Development Program of China under Grant No. 2021YFB3100600, the Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant No. 2021153, and the State Key Program of National Natural Science Foundation of China under Grant No. U2336202.

[*] Corresponding Author

have been proposed[15, 16] to exploit word-level information to enhance character-level Chinese ED. However, these methods experience two problems which are worth further exploration, namely insufficient word-character interaction and event confusing.

For the first problem, interactions between characters and lexicon words are not fully exploited. Specifically, nugget proposal networks (NPNs)[15] first apply a lexicon word list to segment the input sentence into a word sequence and then integrate the word information into character representations with a gate mechanism. Unfortunately, NPNs restrict each character to interact with only one lexicon word, where the one-to-one mapping may lose important word-level semantics to identify the trigger boundary. For example in Fig.1, “讲话” (a speech) triggers a Broadcast-type event. However, the sentence segmentation like S_1 in Fig.1(a) makes the characters “讲” and “话” only receive word-level information that “发表讲话” (deliver a speech) is a whole, which may mislead NPNs to identify “发表讲话” (deliver a speech) as the event trigger instead of “讲话” (a speech). On the contrary, for another example “气象局 (The Meteorological Bureau)/将 (will)/为 (for)/此 (this)/研究 (research)/提供 (provide)/资金支持 (financial support)/”, “资金支持” (financial support) works as a whole to trigger an Transfer-Money-type event. For NPNs, if the sentence is segmented as “气象局 (The Meteorological Bureau)/将 (will)/为 (for)/此 (this)/研究 (research)/提供 (provide)/资金 (finan cial)/支持 (support)/”, it lacks clues to identify “资金支持” (financial support) as a whole, leading to that only “资金” (financial) or “支持” (support) is detected as the trigger. Ding *et al.*[16] proposed Trigger-Aware Lattice Neural Network (TLNN), which constructs cut paths to link the beginning and ending characters. However, the matched words fail to inject the word-level semantics into all the characters they cover except the last one, due to the inherently unidirectional sequential nature of Lattice Long Short-Term Memory (LSTM)[17, 18]. As shown in Fig.1(b), TLNN directly flows the information of “讲话” (deliver a speech) into “话” (speech) but skips “讲” (a), which may lose key clues to identify the trigger boundary.

For the second problem, we observe that it is confusing to discriminate event labels sharing similar semantics. For example, events of “Transfer-Money” and “Transfer-Ownership” usually involve the transfer of money between people or organizations. However, according to the event annotation[1], “Transfer-Money” emphasizes the ownership transfer of money through giving/borrowing instead of purchasing, while “Transfer-Ownership” mainly refers to scenarios about buying and selling. Statistics on Automatic Content Extraction (ACE) 2005, the most widely used Chinese ED dataset, show that 33.3% event labels are semantically confusing. As a result, it is crucial for an ED model to discriminate the confusing event labels. However, few existing Chinese ED methods take this into account.

To address the above problems, this paper presents a novel model, Label-Aware Heterogeneous Graph Attention Network (L-HGAT), for Chinese ED. Specifically, L-HGAT handles the insufficient word-character interaction issue by transforming the input sentence into a heterogeneous graph equipped with two node types and three edge types. As Fig.1(c) shows, each lexicon word connects all the characters it encompasses to achieve adequate information propagation between words and characters, and each character connects its neighboring characters to capture the local context information. Thanks to the hetero-

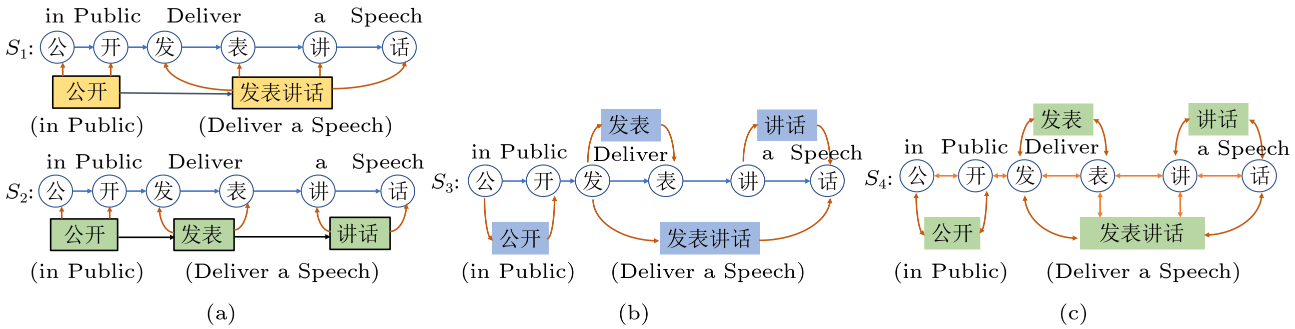


Fig.1. Examples of word-character interaction. (a) Word-character interaction in NPNs. (b) Word-character interaction in Lattice Long Short-Term Memory (LSTM). (c) Our designed word-character interaction.

[1] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf, Nov. 2022.

geneity of different node types and edge types, the heterogeneous graph attentive convolution is performed to aggregate information from neighboring nodes. The event confusing issue is addressed by a newly proposed label-aware matcher. Specifically, we initialize each event label representation with the trigger prototype embeddings to mine the semantic clues. The matcher explicitly teaches the model to distinguish the difference between the ground-truth event types and the confusing counterparts with a margin loss. Compared with previous researches[15, 16, 19], our contributions are as follows.

- To the best of our knowledge, we are the first to adopt a heterogeneous graph to model the word-character interaction, which enhances the character semantics for Chinese ED.
- We design a matcher module to explore the semantic interactions between event triggers and labels, which contributes to discriminating the confusing event labels for Chinese ED.
- Our model consistently achieves superior performances to a range of baselines on two benchmarks, ACE2005[16] and KBP2017[16]. Extensive validation experiments confirm the effectiveness of our model.

2 Related Work

2.1 Chinese Event Detection

Traditional feature-based ED methods[20–27], which heavily rely on hand-crafted features, suffer from the deficiencies in scalability and robustness. In recent years, plenty of studies[6–9, 11–14] have designed neural network models to automatically extract high-level features and achieve great success in English ED. Unfortunately, in Chinese, event triggers are more tricky to identify due to the absence of natural word boundaries. Early methods[28–31] elaborately design hand-crafted features, and some neural network models[15, 16, 19] also incorporate word-level information to enhance the semantics of characters for Chinese ED. However, the Chinese ED researches mentioned above insufficiently explore the word-character interaction. For example, Lin *et al.*[15] ideally limited each character to interacting with only one lexicon word, and Ding *et al.*[16] failed to inject the word-level semantics into all the matched characters. We observe that the sufficient word-character interaction is of great help to recognize event triggers, which motivates us to design new solutions to explore the word-character interaction for Chinese ED.

2.2 Heterogeneous Graph for NLP

The graph neural network[32] was originally designed for homogeneous graphs, where all nodes are of the same type. However, graphs in real scenarios are usually equipped with nodes and edges of multiple types; thus Heterogeneous Graph Neural Network (HetGNN)[33], Heterogeneous Graph Attention Network (HAN)[34] and Heterogeneous Graph Transformer (HGT)[35] were proposed. Motivated by the great performances of HetGNN and HAN in enabling information propagation[36–39], we elaborately design a heterogeneous graph structure to promote the word-character interaction in ED.

2.3 Exploration to Label Semantics

Based on the semantics of target labels, mining the fine-grained matching signals between words and classes has been successfully utilized in the task of text classification[40–42]. Some prior researches[43–45] about ED have tried to explore the semantics of event labels. For example, Huang *et al.*[43] learned event-type representations with event ontology and experimented on the zero-shot scenarios. Further, Lai and Nguyen[44] defined the event type as a set of keywords and focused on discovering new event types. These studies manifest that event label semantics can provide effective signals to boost performances, which motivates us to handle the event confusing problem by mining the semantic clues of event labels.

3 Problem Statement

We formulate Chinese ED as a character-wise sequence labeling paradigm, where each character is assigned a label to indicate whether it is relevant to an event trigger. Specifically, the "begin-inside-other (BIO)"[15] tagging schema is adopted, where the label "O" means that the character is independent of the target event trigger. "B-EventType" and "I-EventType" represent that the character is the beginning and the inside character of an event trigger respectively. "EventType" refers to the specific event type which the trigger evokes. Therefore, the total number of event labels is $2 \times N_{\rm e} + 1$, where $N_{\rm e}$ is the number of predefined event types.

4 Proposed L-HGAT Model

In this section, we introduce the construction of the word-character interactive heterogeneous graph and detail our proposed L-HGAT. Fig.2 shows a toy illustration of the overall model. Specifically, the L-HGAT model consists of three parts. 1) The input layer transforms characters and words into real-valued embeddings and produces their contextualized representations. 2) HGAT layers conduct the message propagation over the graph, enriching the word-character interaction and generating expressive character representations. 3) The label-aware matcher leverages the event label semantics to guide the recognition of event types.

4.1 Graph Construction

For a Chinese sentence $S=\{c_1,c_2,\dots,c_n\}$ containing n characters, its corresponding word sequence is $S_{\rm w}=\{w_1,w_2,\dots,w_m\}$, where $w_i=\{c_{b_i},c_{b_i+1},\dots,c_{e_i-1},c_{e_i}\}$ with b_i and e_i representing the indexes of the beginning and the ending character for the i-th word in S, respectively. For example, in Fig.2, w_1 = “进出口 (The import and export)” with $b_1=1$ pointing to “进 (import)” and $e_1=3$ pointing to “口”. Each sentence is converted to a heterogeneous graph with two types of nodes (characters, words) and three kinds of edges. The heterogeneity of nodes corresponds to the different granularity of semantics of words and characters, and the heterogeneous edges promote functional semantic propagation. The first kind of edges is “c2c-edge”, which connects neighboring characters and captures the contextual character relations to enrich semantics. The second kind of edges is “w2c-edge”, which promotes the information flow between the lexicon word and the corresponding characters, enhancing characters with word-level semantics. The third kind of edges is“c2w-edge”, which is the reverse of “w2c-edge”. The c2w-edges allow information propagation from characters to words, which could alleviate the semantic ambiguity of lexicon words.

4.2 Input Layer (Graph Initializer)

Each character and matched lexicon word are transformed into the corresponding real-valued input embeddings. We denote the input embeddings of S and $S_{\rm w}$ as $\boldsymbol{X}_{\rm c}\in\mathbb{R}^{n\times d}$ and $\boldsymbol{H}_{\rm w}=(\boldsymbol{h}^0_{1,\rm w},\boldsymbol{h}^0_{2,\rm w},\dots,\boldsymbol{h}^0_{m,\rm w})\in\mathbb{R}^{m\times d}$respectively, where d is the hidden size, n and m are the number of characters and matched words in the sentence respectively. With $\boldsymbol{X}_{\rm c}\in\mathbb{R}^{n\times d}$ as input, a basic encoder is adopted to produce expressive character representations $\boldsymbol{H}_{\rm c}=(\boldsymbol{h}^0_{1,\rm c},\boldsymbol{h}^0_{2,\rm c},\dots,\boldsymbol{h}^0_{n,\rm c}\in\mathbb{R}^{n\times d})$ in the sentence. Subsequently, $\boldsymbol{H}_{\rm c}$ and $\boldsymbol{H}_{\rm w}$ are used as initial node features in the following HGAT layers.

4.3 HGAT Layer

We leverage the Heterogeneous Graph Attention Network (HGAT)[34] to enable the word-character semantic propagation over the graph. Specifically, the node-level attention enhanced graph convolution is first adopted to produce the initial semantic embedding for each node. Further, for character-type nodes, the type-level attention fuses the semantic embeddings from different types of neighboring nodes. L layers of HGAT are used to produce the final character representations.

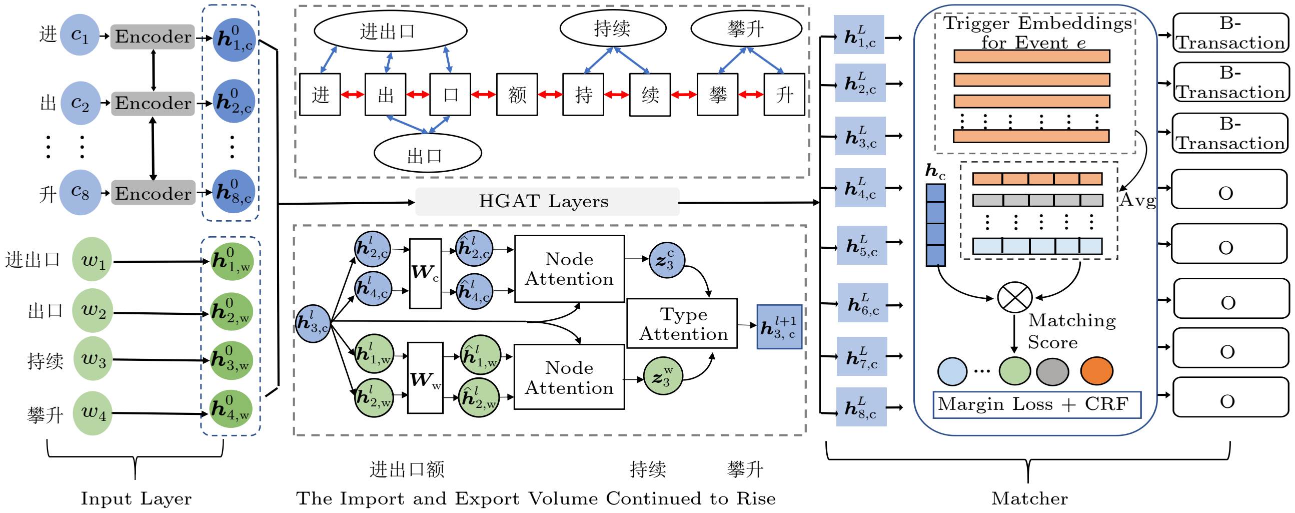

进 c_1 Encoder $\boldsymbol{h}^0_{1,\rm c}$
出 c_2 Encoder $\boldsymbol{h}^0_{2,\rm c}$
升 c_8 Encoder $\boldsymbol{h}^0_{8,\rm c}$
进出口 w_1 $\boldsymbol{h}^0_{1,\rm w}$
出口 w_2 $\boldsymbol{h}^0_{2,\rm w}$
持续 w_3 $\boldsymbol{h}^0_{3,\rm w}$
攀升 w_4 $\boldsymbol{h}^0_{4,\rm w}$
进出口
持续
攀升
进 出 口 额 持 续 攀 升
出口
HGAT Layers
$\boldsymbol{h}^l_{3,\rm c}$
$\boldsymbol{h}^l_{2,\rm c}$ $\boldsymbol{h}^l_{4,\rm c}$ $\boldsymbol{W}_{\rm c}$ $\hat{\boldsymbol{h}}^l_{2,\rm c}$ $\hat{\boldsymbol{h}}^l_{4,\rm c}$
Node Attention
$\boldsymbol{z}^{\rm c}_3$
$\boldsymbol{h}^l_{1,\rm w}$ $\boldsymbol{h}^l_{2,\rm w}$ $\boldsymbol{W}_{\rm w}$ $\hat{\boldsymbol{h}}^l_{1,\rm w}$ $\hat{\boldsymbol{h}}^l_{2,\rm w}$
Node Attention
$\boldsymbol{z}^{\rm w}_3$
Type Attention
$\boldsymbol{h}^{l+1}_{3,\rm c}$
$\boldsymbol{h}^L_{1,\rm c}$ $\boldsymbol{h}^L_{2,\rm c}$ $\boldsymbol{h}^L_{3,\rm c}$ $\boldsymbol{h}^L_{4,\rm c}$ $\boldsymbol{h}^L_{5,\rm c}$ $\boldsymbol{h}^L_{6,\rm c}$ $\boldsymbol{h}^L_{7,\rm c}$ $\boldsymbol{h}^L_{8,\rm c}$
Trigger Embeddings for Event e
$\boldsymbol{h}_{\rm c}$
Avg
Matching Score
Margin Loss + CRF
B-Transaction
B-Transaction
B-Transaction
O
O
O
O
O
Input Layer
进出口额 持续 攀升
The Import and Export Volume Continued to Rise
Matcher

Fig.2. Toy illustration of our proposed L-HGAT model.

4.3.1 Node Attention

The attentive graph convolution[46] works to aggregate features from neighboring nodes, where the node-level attention mechanism measures how neighboring nodes impact each other. Due to the heterogeneity of nodes[34], we introduce two node-type specific transformation matrices $\boldsymbol{W}_\tau$ with $\tau \in \{w, c\}$ to project word-type and character-type node features respectively. In the l-th HGAT layer, the project process for a τ-type node j is shown as follows:

$$\hat{\boldsymbol{h}}^l_{j,\tau} = \boldsymbol{W}_\tau \boldsymbol{h}^l_{j,\tau},$$

where $\boldsymbol{W}_\tau \in \mathbb{R}^{d\times d}$ is the node-type specific convolution filter, and $\boldsymbol{h}^l_{j,\tau}$ and $\hat{\boldsymbol{h}}^l_{j,\tau}$ are the original and projected feature representations of node j, respectively.

Further, for a τ'-type node i, the attentive graph convolution is exploited to each type of its neighboring nodes individually, producing the initial semantic embeddings for node i as follows:

$$e_{ij} = \mathrm{LeakyReLU}((\boldsymbol{v}_\tau)^{\mathrm{T}}[\hat{\boldsymbol{h}}^l_{i,\tau'}, \hat{\boldsymbol{h}}^l_{j,\tau}]),$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum\limits_{j\in N_{\tau,i}} \exp(e_{ij})},$$

$$\boldsymbol{z}^\tau_i = \sigma\left(\sum_{j\in N_{\tau,i}}^{n} a_{ij}\hat{\boldsymbol{h}}^l_{j,\tau}\right),$$

where $\boldsymbol{v}_\tau \in \mathbb{R}^{(2\times d)\times 1}$ is a trainable vector for attention computation and LeakyReLU is the activation function. $\boldsymbol{z}^\tau_i$ is the semantic embedding aggregated from τ-type neighbors. $N_{\tau,i}$ is the set of τ-type neighbors of node i, and a_{ij} is the attention coefficient indicating the importance of node j to node i. Please notice that a_{ij} is asymmetric since node i and node j differently impact each other. In our heterogeneous graph, there are three update processes owing to the three types of edges. Specifically, τ and τ' are the character-type and word-type along c2w-edges, character-type and character-type along c2c-edges, and word-type and character-type along w2c-edges.

4.3.2 Type Attention

For a character node i, we obtain two types of semantic embeddings, $\boldsymbol{z}^{\mathrm{c}}_i$ and $\boldsymbol{z}^{\mathrm{w}}_i$, from its neighboring character and word nodes respectively. To output a comprehensive representation $\boldsymbol{h}^{l+1}_{i,\mathrm{c}}$ for the next layer, type-attention is utilized to fuse these two semantic embeddings. Specifically, we weight these semantic embeddings as follows:

$$w_{i,\tau} = \frac{1}{|\{w, c\}|}((\boldsymbol{q})^{\mathrm{T}}\tanh(\boldsymbol{W}_t\boldsymbol{z}^\tau_i + \boldsymbol{b})),$$

$$\beta_{i,\tau} = \frac{\exp(w_{i,\tau})}{\sum\limits_{\tau\in\{w,c\}} \exp(w_{i,\tau})},$$

where $\beta_{i,\tau}$ is a scalar measuring the importance of τ-type neighbors to node i, and $\boldsymbol{W}_t$, $\boldsymbol{b}$, $\boldsymbol{q}$ are trainable model parameters. The learned coefficients highlight the valuable semantic embeddings, and we fuse the semantic embeddings for character-type nodes as follows:

$$\boldsymbol{h}^{l+1}_{i,\mathrm{c}} = \sum_{\tau\in\{w,c\}} \beta_{i,\tau}\boldsymbol{z}^\tau_i,$$

where $\boldsymbol{h}^{l+1}_{i,\mathrm{c}}$ is the representation for the next HGAT layer. For word-type nodes whose neighboring nodes are only of character-type, the learned semantic embeddings are used as representations for the next layer, namely $\boldsymbol{h}^{l+1}_{i,\mathrm{w}} = \boldsymbol{z}^{\mathrm{c}}_i$.

4.4 Label-Aware Matcher

We design a label-aware matcher to guide the discrimination to confusing event labels, where three key steps are involved.

The first step is to derive the representations of event labels. In this step, we first convert each event label into a real-valued embedding. Since event labels are normally semantically related to the corresponding event triggers, we exploit the trigger-prototype character embeddings as the initial representations of event labels. Concretely, in the data pre-processing phase, we gather all event trigger characters for each event label in the training set and convert them into real-valued embeddings. For example, supposing that the k-th event label, "B-Attack", contains l trigger characters $\{t_{k1}, t_{k2}, \ldots, t_{kl}\}$, we embed these characters by looking up the pretrained character embeddings. Then, we utilize the average vector of these embeddings as the initial embedding of label "B-Attack". So does the operation for other event labels, and we formulate this process as follows:

$$\boldsymbol{E}_k = \frac{1}{l}\sum_{j=1}^{l} \boldsymbol{e}(t_{kj}),$$

where $\boldsymbol{E}_k$ is the embedding of the k-th event label, t_{kj} denotes the j-th trigger character for the k-th event label, and $\boldsymbol{e}(t_{kj})$ is the operation of looking up the character embeddings. All the trigger-prototype label embeddings are concatenated to form $\boldsymbol{E}=(\boldsymbol{E}_1, \boldsymbol{E}_2, \dots, \boldsymbol{E}_k, \dots, \boldsymbol{E}_{(2\times N_\mathrm{e}+1)})\in\mathbb{R}^{(2\times N_\mathrm{e}+1)\times d}$ as the initialized label embedding matrix, where $2\times N_\mathrm{e}+1$ is the number of event labels mentioned in Section 3, and d is the dimension of label embedding.

The second step is to compute the matching scores. For each character c_i in the sentence, we compute a matching score vector between it and all event labels. The score measures which type of event trigger c_i is most likely to be. Given c_i's representation $\boldsymbol{h}_{i,c}^L\in\mathbb{R}^d$ from the last HGAT layer, we use dot product with $\boldsymbol{E}\in\mathbb{R}^{(2\times N_\mathrm{e}+1)\times d}$ to compute c_i's matching score vector $\boldsymbol{s}_{c_i}\in\mathbb{R}^{(2\times N_\mathrm{e}+1)}$ as follows:

$$\begin{aligned}\boldsymbol{s}_{c_i}&=\boldsymbol{h}_{i,\mathrm{c}}^L\boldsymbol{E}^\mathrm{T}\\&=(\boldsymbol{h}_{i,\mathrm{c}}^L\boldsymbol{E}_1,\ \dots,\ \boldsymbol{h}_{i,\mathrm{c}}^L\boldsymbol{E}_k,\ \dots,\ \boldsymbol{h}_{i,\mathrm{c}}^L\boldsymbol{E}_{2\times N_\mathrm{e}+1})\\&=(s_i^1,\ \dots,\ s_i^k,\ \dots,\ s_i^{2\times N_\mathrm{e}+1}),\end{aligned}$$

where s_i^k is a scalar representing the matching score between the character c_i and the k-th event label.

The third step is to discriminate the event labels. Assuming that the ground-truth event label for c_i is the k-th label, then s_i^k is the corresponding matching score. For other matching scores except s_i^k, the highest one is for the most confusing event label, and we denote the label as $s_i^{\hat{k}}$. In the ideal situation, we would have $s_i^k > s_i^{\hat{k}}$, which means that it is easy for the model to recognize the correct event label. However, some confusing event labels are likely to get higher matching scores than the correct label; hence margin loss is leveraged to penalize our architecture as follows:

$$L_m(c_i)=\max(m+s_i^{\hat{k}}-s_i^k,0),$$

where m is a positive margin. This loss function could penalize the architecture even when $s_i^k > s_i^{\hat{k}}$ but the gap between them is not large enough. The pushing-away game of the margin loss drives the model to discriminate confusing event labels better.

4.5 Training and Inference

During training, two kinds of losses are involved. Specifically, the first kind of loss is conditional random field (CRF) loss from sequence tagging. Since we cast ED as a sequence labeling problem, CRF is employed as a sequence tagger using the aforementioned computed matching scores as inputs. For each sentence $S=\{c_1,c_2,\dots,c_n\}$, which contains a corresponding label sequence $L=\{y_1,y_2,\dots,y_n\}$ and the matching score vector $\boldsymbol{s}_{c_i}\in\mathbb{R}^{(2\times N_\mathrm{e}+1)}$ for each character, the probability of L is derived as follows:

$$P(L|S)=\frac{\exp\left(\sum\limits_{i=1}^{n}(\boldsymbol{s}_{c_i}+T(y_{i-1},y_i))\right)}{\sum\limits_{L'\in\mathbb{C}}\exp\left(\sum\limits_{i=1}^{n}(\boldsymbol{s}'_{c_i}+T(y'_{i-1},y'_i))\right)},$$

where $\mathbb{C}$ is the set of all arbitrary label sequences, and $T(y'_{i-1},y'_i)$ is the transition function to compute the transition score between (y_{i-1},y_i). We use the viterbi[47] algorithm to decode the highest scored label sequence, and get the CRF loss as follows:

$$L_\mathrm{crf}=-\log(P(L|S)).$$

The second kind of loss is margin loss from the label-aware matcher. The sentence-level margin loss is obtained by summing the margin losses of each character c_i in the sentence, and we specify it as follows:

$$L_m=\sum_{i=1}^{n}L_m(c_i).$$

The final optimization objective for sentence S is:

$$L=L_\mathrm{crf}+\alpha L_m,$$

where α is a hyper-parameter, which controls the relative impact of margin loss and exponentially decays with each training epoch.

During the model inference, we first obtain the character representations from the last layer of HGAT. Then, the label-aware matcher computes the matching scores between each character and the label embeddings. Finally, the viterbi algorithm infers the highest scored label sequence to tag the event triggers.

5 Experiments

5.1 Experimental Settings

We detail our experimental settings from three aspects.

1) *Datasets*. We conduct experiments on two popular datasets, ACE2005[48] and TAC KBP 2017 Event Nugget Detection Evaluation Dataset (KBP2017)[2]. For a fair comparison, we use the same data splits as the previous study[49]. For ACE2005, 569/64/64 articles are used as the training/validation/test set. For KBP2017, we use 506/20/167 documents as the train-

ing/validation/test set. These two datasets have 26 event types in common. We deploy these confusing event labels in each dataset in Table 1.

2) *Evaluation*. Following previous studies[15, 16], we use micro-averaged precision, recall and F_1 as evaluation metrics for ACE2005, and use the official evaluation toolkit[2] for KBP2017 to obtain these metrics. An event trigger is correctly classified when its boundary and event type simultaneously meet the gold answer.

3) *Implementation Details*. We manually tune the hyper-parameters on the validation set. We try two encoders in this paper. The first encoder is a 1-layer bidirectional LSTM (BiLSTM), where the hidden size of the whole L-HGAT is 100. The model is optimized via stochastic gradient descent with the learning rate of 0.1. Further, L2 regularization with a parameter of 1×10^{-5} is used to avoid overfitting. Another encoder is bidirectional encoder representation from transformers (BERT)[50], where we use the base Chinese model[3]. In this situation, the model is optimized using the Adam[51] optimizer with the learning rate of 2×10^{-5}. The number of HGAT layers is 2. The maximum length of the sentence is set to 250. We use the same lexicon word list as previous studies[15, 16] to obtain the word boundaries. We run all experiments using PyTorch 1.5.1 with Python3.7 on the NVIDIA Tesla T4 GPU.

5.2 Baselines

To comprehensively evaluate our L-HGAT model, we compare it with a series of baselines, which could be divided into four categories.

Feature-based models make use of human-designed features to conduct ED. 1) Rich features for Chinese EE (Rich-C*)[52] utilize hand-crafted linguistic features. 2) CLUZH (KBP2017 Best)[53] incorporates heuristic features into the encoder and achieves the best performance in KBP2017.

Character-based models formulate Chinese ED as a character-level multi-class classification problem or sequence labeling problem. 1) Dynamic Multi-Pooling Convolutional Neural Network (DMCNN)[6] uses dynamic multi-pooling convolution to learn character-wise features for ED. 2) Convolution Bilstm Neural Network (C-LSTM)[54] combines LSTM and the convolutional neural network to capture the sentence-level semantics for ED. 3) Hierarchical and Bias Tagging Network with Gated Multi-Level Attention Mechanisms (HBTNGMA)[12] integrates the character-wise sentence and document information through a hierarchical and bias tagging network for ED.

Word-based models convert Chinese ED into a word-level multi-class classification problem or sequence labeling problem, where a lexicon word list is utilized to get word boundaries in the data pre-processing process. 1) The model architectures of DMCNN_word[6] and HBTNGMA_word[12] are the same as those of character-based models, but they are employed in the level of words. 2) Hybrid Neural Network (HNN)[49] combines features extracted from CNN with BiLSTM to perform ED.

Table 1. Confusing Event Label Pairs

Event Pair	Dataset	Explanation
(Die, Injure)	ACE2005, KBP2017	(Die, Injure) frequently occurs in similar contexts since Injure could result in Die
(Charge-Indict, Sue)	ACE2005, KBP2017	(Charge-Indict, Sue) refers to that a person/organization is accused of a crime, but Charge-Indict emphasizes that the plaintiff should be a state actor
(Transfer-Money, Transfer-Ownership)	ACE2005, KBP2017	Transfer-money refers to the ownership transfer of money via giving/borrowing, while Transfer-Ownership refers to the ownership transfer via buying/selling
(Sue, Appeal)	ACE2005, KBP2017	(Sue, Appeal) both refer to raising lawsuit to the court, but Appeal occurs when the lawsuit is taken to a higher court for review
(Sentence, Extradite)	ACE2005	(Sentence, Extradite) relates to the punishment announced by a state actor, but Extradite means that the person is sent to another country for punishment
(Acquit, Pardon)	ACE2005	(Acquit, Pardon) both refer to that the accused will not be punished, but the former emphasizes innocence while the latter emphasizes excuse
(Contact, Meet)	KBP2017	(Contact, Meet) refer to the communication among people, but Meet emphasizes that the communication happens in a face-to-face manner, while Contact also includes communications through letter/phone, etc.
(Transaction, Transfer-Money)	KBP2017	Transaction depicts the import/export between countries with money transferred, while Transfer-Money refers to the money transfer via giving/borrowing

[2] https://github.com/hunterhector/EvmEval, Sept. 2022.
[3] https://huggingface.co/bert-base-chinese, Sept. 2022.

Hybrid models conduct character-wise Chinese ED with lexicon word information. 1) NPNs[15] exploit character compositional structures of event triggers and utilize the gate mechanism to summarize the information from the character and word sequence. 2) TLNN[16] utilizes trigger-aware lattice neural networks enhanced with semantics from external linguistic knowledge bases[4]. 3) Hybrid Character Representation (HCR)[19] incorporates word information and pre-trained language models to improve the character-wise BiLSTM+CRF model. HCR(BERT) achieves the best performance on ACE2005.

5.3 Overall Results

Table 2 and Table 3 summarize the performances of our proposed model[5] and other baselines on the two datasets. Note that italics are results with BERT, numbers in bold are the best results of the corresponding evaluation indicator, and numbers in underline are the best results without BERT. Experiments show that our proposed L-HGAT exceeds baseline methods on both benchmarks, and we have observations as follows.

1) Sufficient word-character interaction is of great importance to boosting performances. Table 2 and Table 3 both reflect that the word-based models outperform character-based ones, which directly proves the importance of word-level information. Further, hybrid models surpass both character-based and word-based models by a large margin, which provides

Table 2. Experimental Results on ACE2005 Dataset

Category	Model	Trigger Identification			Trigger Classification		
		Precision	Recall	F_1	Precision	Recall	F_1
Feature	Rich-C* [52]	62.20	71.90	66.70	58.90	68.10	63.20
Character	DMCNN[6]	60.10	61.60	60.90	57.10	58.50	57.80
	C-LSTM[54]	65.60	66.70	66.10	60.00	60.90	60.40
	HBTNGMA[12]	41.67	59.29	48.94	38.74	55.13	45.50
Word	DMCNN_word[6]	66.60	63.60	65.10	61.60	58.80	60.20
	HNN[49]	**74.20**	63.10	68.20	**77.10**	53.10	63.00
	HBTNGMA_word[12]	54.29	62.82	58.25	49.86	57.69	53.49
Hybrid	HCR[19]	60.30	73.30	66.20	58.10	70.60	63.70
	NPNs[15]	70.63	64.74	67.56	67.13	61.54	64.21
	TLNN[16]	67.39	68.91	68.14	64.57	66.02	65.29
	HCR(BERT)	*68.90*	***78.80***	*73.50*	*66.40*	***76.00***	*70.90*
Ours	HGAT	68.20	71.47	69.80	64.22	67.30	65.73
	L-HGAT	71.99	70.83	71.41	69.38	68.27	68.82
	L-HGAT(BERT)	*73.07*	*75.64*	***74.33***	*70.28*	*72.76*	***71.49***

Table 3. Experimental Results on KBP2017 Dataset

Category	Model	Trigger Identification			Trigger Classification		
		Precision	Recall	F_1	Precision	Recall	F_1
Feature	CLUZH(KBP2017 Best)[53]	**67.76**	45.92	54.74	62.69	42.48	50.64
Character	DMCNN[6]	53.67	49.92	51.73	50.03	46.53	48.22
	HBTNGMA[12]	40.52	46.76	43.41	35.93	41.47	38.50
Word	DMCNN_word[6]	53.67	49.92	51.73	50.03	46.53	48.22
	HBTNGMA_word[12]	46.92	53.57	50.02	37.54	42.86	40.03
Hybrid	NPNs[19]	58.03	59.91	58.96	52.04	53.73	52.87
	TLNN[16]	60.50	56.79	58.59	59.23	53.11	56.00
Ours	HGAT	61.90	**62.84**	62.37	56.48	**57.34**	56.90
	L-HGAT	63.91	60.06	61.92	59.21	55.64	57.37
	L-HGAT(BERT)	*65.85*	*59.51*	***62.52***	***61.62***	*55.68*	***58.50***

[4] Since we could not acquire the external sense embedding used in TLNN, we reproduce TLNN with the same Glove embedding used in this paper for a fair comparison.

[5] HGAT works without the label-aware matcher, where the character representations produced by L layers of HGAT are followed by a fully-connected layer and CRF.

strong evidence for the superiority of integrating semantics from characters and matched lexicon words. As far as our proposed model, it exceeds all baselines. We attribute such improvement to that the heterogeneous graph-based interaction brings rich semantic propagation between words and characters.

2) The label-aware matcher, which incorporates the semantics of event labels and employs a margin loss, provides signals to predict event triggers precisely. We notice that HGAT and L-HGAT show advantages on different evaluation indicators. HGAT mainly advances on the indicator of recall, since more potential event triggers are detected with the adequate word-character information propagation. Meanwhile, thanks to the label-aware matcher, L-HGAT improves the performance on the indicator of precision, where triggers are predicted more precisely.

3) Pre-trained language models contribute to improving the overall performances. Following Xi *et al.*[19], we also utilize BERT to boost performances. The noticeable performance improvement on both datasets demonstrates the effectiveness of pre-trained language models.

6 Analysis and Discussion

6.1 Ablation to Heterogeneous Graph

We conduct ablation studies and report the final performances of trigger classification (F_1). We provide detailed results in Table 4 and Table 5, where "w/o" is short for "without".

To probe the reasonability of our designed word-character interactive graph, we ablate each component of it. Reading from the results in Table 4, we have observations as follows. 1) The ablation to c2c-edges brings grievous damage to F_1, which demonstrates the necessity of local context information between characters. 2) The removal of all w2c-edges limits the word information to flowing into only the last character as lattice-LSTM does. Hence, the word-character dependency is insufficiently explored and the results drop. 3) The performance degradation after removing c2w-edges demonstrates that the lexicon words may introduce noisy information. This ablation verifies that c2w-edges could inject contextual sentence information into the words to rectify the corresponding word-level semantics. 4) We also remove all the word nodes, making the heterogeneous graph degenerate into a character-level homogeneous one. The obvious performance drop in results verifies that incorporating word-level semantics is important. Besides, L-HGAT w/o word still exceeds other character-based-level baselines in Table 2 and Table 3, which shows the effectiveness of local context and the label-aware matcher.

Table 4. Ablation to Heterogeneous Graph

Model	Trigger Classification (F_1)	
	ACE2005	KBP2017
L-HGAT	68.82	57.37
w/o c2c-edges	65.13	55.26
w/o full w2c-edges	66.35	55.07
w/o c2w-edges	65.33	56.07
w/o word	64.54	52.27

Table 5. Ablation to Graph Embedding Strategy

Model	Trigger Classification (F_1)	
	ACE2005	KBP2017
L-HGAT	68.82	57.37
w/o $\boldsymbol{W}_\tau$	66.81	55.93
w/o type-attnention	64.51	56.50
L-HetGNN	66.33	56.52
L-HGT	65.81	56.15

Further, we ablate to HGAT and try different graph embedding strategies to compare the performances. Reading from Table 5, we have analysis as follows.

1) We replace the node type-specific convolution filter as a universe one. The performance degradation indicates that the node type-specific convolution filter, which serves to capture different granularities of semantics provided by words and characters, is indispensable.

2) L-HGAT w/o type-attention derives the node representation by directly adding the semantic embeddings aggregated from each type of neighboring nodes. The type-attention operation could highlight the important semantics from words and characters, while the rough "add" operation ignores to refine the valuable semantics. The performance degradation confirms the importance of choosing an appropriate approach for semantic fusion.

3) To probe how the graph embedding strategy influences the final results, we additionally choose HetGNN[33] and HGT[35] as variants. For HetGNN, we sample four heterogeneous neighbors for each node. For HGT, we attempt different numbers of heads in heterogeneous mutual attention computation where the results peak with four heads. Armed with the label-aware matcher, we denote these variant models as L-HetGNN and L-HGT respectively. We report the

results in Table 5 and derive the following observations. 1) The change of graph embedding strategy impacts the final performances, but all the variants of embedding strategy excel the previous baselines. 2) Though the performances drop with the change of graph embedding strategy, such ablation hurts performance less than the variants of graph construction in Table 4. These two observations confirm that graph construction over-weights the embedding strategy, which further verifies the contribution of our designed word-character interaction graph.

Last but not the least, we notice that all variants above hurt performances worse in ACE2005 than in KBP2017. According to Ding *et al.*[16], 85.39% event triggers in ACE2005 are exactly lexicon words while the percentage in KBP2017 is 76.28%. This explains why the insufficient or unreasonable word-character interaction hurts performances in ACE2005 more severely. For example, “L-HGAT w/o type-attention” and “L-HGAT w/o c2w-edges” do not filter noisy semantics from matched lexicon words, and these two ablations impact performance in ACE2005 worse than that in KBP2017.

6.2 Ablation to the Label-Aware Matcher

We also ablate the label-aware matcher to probe how it contributes to the final performances. Table 6 reflects that L-HGAT achieves substantial improvement over HGAT, which demonstrates that the label-aware matcher is able to provide fine-grained semantic signals to benefit ED. Further, we have analysis as follows.

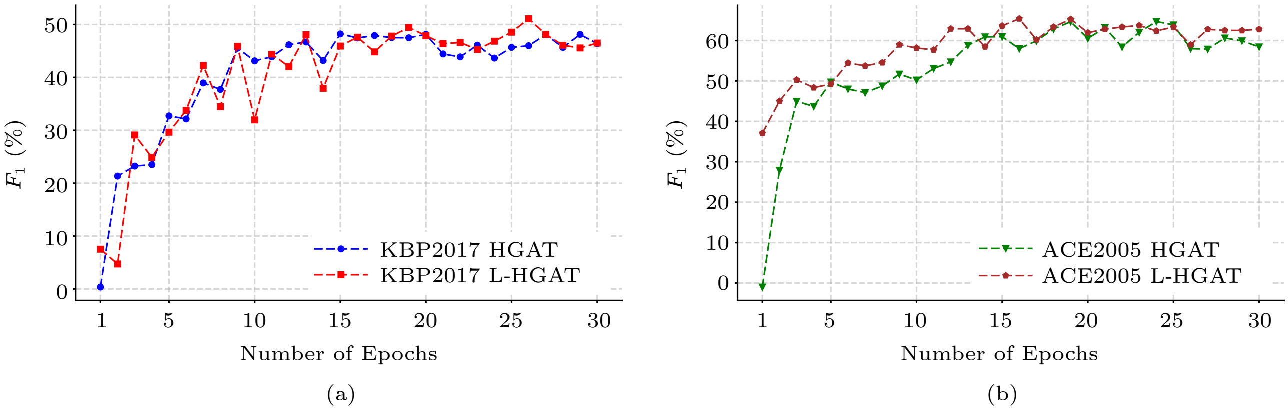
1) We first probe the training process of L-HGAT and HGAT. As Fig.3 illustrates, L-HGAT remarkably surpasses HGAT on the validation set in the early training stage and maintains its advantage in the whole training process. This phenomenon reveals that the semantics of event labels could provide prior knowledge to ease the training process, guiding the detection of event triggers.

Table 6. Ablation to Label-Aware Matcher

Model	Trigger Classification (F_1)	
	ACE2005	KBP2017
L-HGAT	68.82	57.37
HGAT	65.73	56.90
L-HGAT w/o margin loss	65.56	55.79
L-HGAT w/o label embeddings	63.93	56.20

2) Table 6 shows that the individual removing to the label embeddings or margin loss could result in worse performances than removing them simultaneously. We infer the reasons for such a phenomenon from two aspects. On the one hand, trigger-prototype-based label embeddings provide prior semantic signals to event labels, but the confusing semantics is also introduced. In this situation, it is necessary to employ the margin loss to lower the matching score computed with the most confusing label, where the ability to discriminate event labels is enhanced. On the other hand, without the trigger-prototype-based label embeddings, the randomly initialized embedding matrix contains no semantic information; thus the margin loss may mislead the matching scores between characters and the corresponding event labels.

3) We investigate why the performances in ACE2005 are worse damaged than in KBP2017 with three matcher variants. We count the proportion of confusing label samples. Statistics show that 28.8% samples in the test set of ACE2005 suffer from the event confusing problem while the proportion is 24.1% in that of KBP2017. Such statistics reveal that ACE2005 is more severely ill-equipped to the event confusing problem; thus the performances of the laebl-



(a) (b)

Fig.3. F_1 variation with training epochs. (a) Results on the KBP2017 validation set. (b) Results on the ACE2005 validation set.

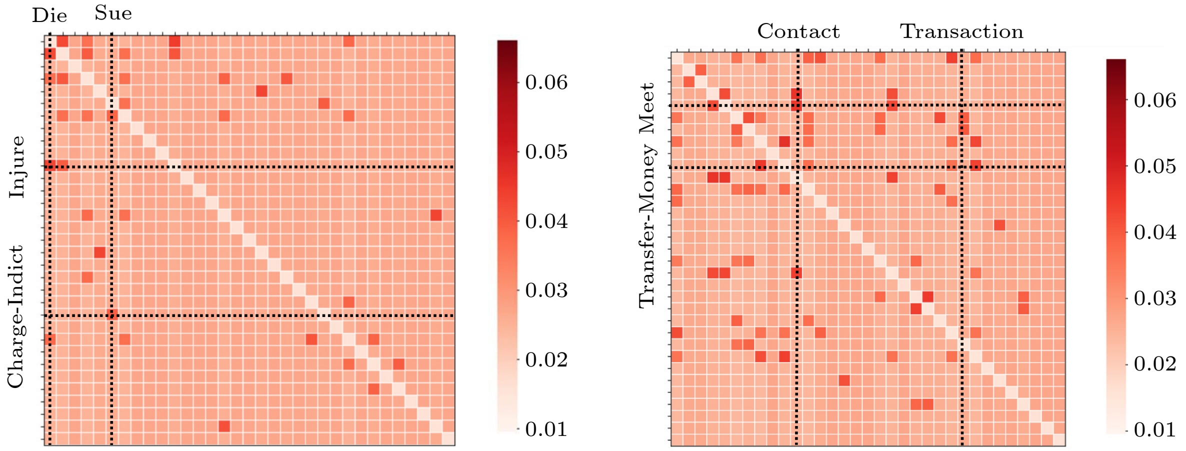
aware matcher variants in ACE2005 fluctuate more violently.

6.3 Interpretability of Label Embedding

The label-aware matcher initializes label embeddings using trigger-prototype-based embeddings and fine-tunes the embeddings during training. To investigate whether the fine-tuned event label embeddings capture the difference and relevance between event labels, we calculate the similarity between each pair of the confusing event labels. Specifically, since each trigger must contain a "B-EventLabel" which marks its beginning character, we use the embedding of "B-EventLabel" to derive the corresponding event label representations. We calculate the cosine similarity between event labels, and present the visualization results as shown in Fig.4. For clarity, we mask the diagonal score to eliminate meaningless self-similarity, and use the softmax function to normalize the score of each line. Fig.4 shows that the similarity matrix is very sparse, since most event labels are semantically discriminative to each other. We further notice that some event labels carry relatively high similarity values, and these event labels share similar semantics with each other, such as (Die, Injure) and (Meet, Contact). Therefore, we believe that label embedding is interpretable and capable of providing semantic clues for ED.

6.4 Discrimination to Confusing Event Labels

To measure whether the label-aware matcher facilitates the event label discrimination, we count the trigger classification (F_1) of HGAT and L-HGAT on confusing event label pairs and present the results in Table 7 and Table 8. Specifically, since different types of event triggers are surrounded by contexts with different semantics, they could push the corresponding event labels to differ with each other during training. We take the event pair (Charge-Indict, Sue) as an example. The contexts of "Charge-Indict" usually contain a state organization/actor (i.e., these criminals are indicted by the police), while the contexts of "Sue" is usually a normal person (i.e., if you do not complete the work, I will sue you for damages). Dif-



(a)

(b)

Fig.4. Visualization of label embedding similarity (row-wise normalized), where we only mark some typical event label pairs for better visualization effect due to the space limitation. (a) Visualization to ACE2005. (b) Visualization to KBP2017.

Table 7. Performance on Confusing Event Labels in ACE2005

Event Pair	Label	F_1		Similarity Reduction Δ (%)
		HGAT	L-HGAT	
(Injure, Die)	Injure	83.4	87.2	+3.8
	Die	71.2	72.4	+1.2
(Transfer-Money, Transfer-Ownership)	Transfer-Money	28.1	26.6	−1.5
	Transfer-Ownership	45.9	49.4	+3.5
(Meet, Contact)	Meet	46.0	45.2	−0.8
	Contact	8.6	13.2	+4.6
(Transaction, Transfer-Money)	Transaction	0.0	3.2	+3.2
	Transfer-Money	28.1	26.6	+1.5

Table 8. Performance on Confusing Event Labels in KBP2017

Event Pair	Label	F_1		Similarity Reduction Δ (%)
		HGAT	L-HGAT	
(Injure, Die)	Injure	1.0	1.0	0.0
	Die	77.4	77.4	0.0
(Charge-Indict, Sue)	Charge-Indict	72.7	88.9	+16.2
	Sue	52.6	60.0	+7.4
(Transfer-Money, Transfer-Ownership)	Transfer-Money	71.4	80.0	+8.6
	Transfer-Ownership	78.2	78.3	+0.1
(Sue, Appeal)	Sue	52.6	60.0	+7.4
	Appeal	73.6	77.8	+4.2

ferent contexts inject characteristic semantics into the trigger characters. Accordingly, to map the event labels with the corresponding triggers, the event label representations are fine-tuned adaptly with the trigger semantics. Therefore, the discriminative information between labels is learned.

Note that the label-aware matcher sometimes may be too strict to assign event labels, which may slightly hurt performances. For example, although the label-aware matcher correctly predicts the Contact-type triggers which are wrongly classified as Meet, some Meet-type triggers are not detected due to the overly strict semantic distinction. This could explain the slightly performance drop on Meet-type and Transfer-Money-type triggers. Besides, such a phenomenon also confirms that the performance improvement from HGAT to L-HGAT mainly comes from the precision indicator.

Further, we calculate the similarity of semantically confusing event labels before and after fine-tuning. From Table 9, we notice that the similarity between confusing event labels obviously reduces. The similarity reduction provides fine-grained semantic clues to discriminate confusing event labels, which helps to predict the event type of triggers more precisely.

Table 9. Similarity Reduction of Confusing Event Labels

Event Label Pair	Similarity Reduction Δ (%)
(Pardon, Acquit)	–76.0
(Die, Injure)	–6.0
(Sentence, Extradite)	–14.0
(Sue, Charge-Indict)	–5.8
(Meet, Contact)	–11.5
(Sue, Appeal)	–5.2
(Transfer-Ownership, Transfer-Money)	–6.4

6.5 Discussion About Trigger Mismatch

The trigger-word mismatch problem, where an event trigger could be part of a lexicon word or contain multiple words, is a typical phenomenon in Chinese ED. Such mismatch between triggers and lexicon words raises the difficulty of precisely locating event triggers. Following previous studies[15, 16], to probe how the trigger-word mismatch problem is alleviated with our model, we also count the recall rate of mismatch triggers in the trigger identification stage. For a fair comparison, we analyze the performance of HGAT, since L-HGAT introduces the semantics of event labels which is not leveraged in previous studies. Table 10 signifies that HGAT handles the word-trigger mismatch better than our chosen baselines, which verifies that adequate word-character exploration could bring improvement to trigger identification.

Table 10. Recall of Trigger-Word Mismatch Triggers on the Test Set of Both Datasets in Trigger Identification Stage

Model	ACE2005	KBP2017
HGAT	92.30	73.63
NPNs	84.61	64.55
TLNN	61.53	63.63

6.6 Case Study

Table 11 shows examples about the word-character interaction problem. In the first case, a word, which consists of two sub-words “交换” (exchange) and “意见” (views), triggers a Meet-type event. TLNN predicts the second sub-word “意见” (views) as the event trigger, since “见” receives the most lexicon features. Meanwhile, NPNs segment the sentence as “两国 (The two countries)/ 代表 (representatives)/ 深入 (in depth)/ 交换 (exchange)/ 意见 (views)”, where the semantics of the complete word could not be aware. Hence, NPNs fail to correctly identify the trigger. In contrast, thanks to the sufficient word-character interac-

Table 11. Case Study for Word-Character Interaction

Model	Case 1: 两国代表深入交换意见... Representatives of the two countries exchanged views in-depth.	Case 2: 市政府对广场实施了大规模 改造工程... The municipal government has implemented a large-scale reconstruction project for the square.
NPNs	(交换, Meet)	(改造工程, Artifact)
TLNN	(意见, Meet)	(改造工程, Artifact)
HGAT	(交换意见, Meet)	(改造, Artifact)
L-HGAT	(交换意见, Meet)	(改造, Artifact)
Answer	(交换意见, Meet)	(改造, Artifact)

tion, our model accurately detects the trigger “交换意见” (exchange views). For the second case, since NPNs segment “改造工程” (reconstruction project) as a whole, the semantics of “改造” (reconstruction) could not be captured and thus the wrong prediction is made.

Table 12 gives two examples of how the label-aware matcher contributes to more precise predictions. Reading from Fig.4, “Transaction” and “Transfer-Money” are semantically similar since they both involve the behaviors of transferring money. HGAT predicts “出口” (export) as “Transfer-Money” while L-HGAT correctly predicts it as “Transaction”. We owe the excellent performances of L-HGAT to the label-aware matcher, where the label embeddings guide the recognition of event labels and the margin loss helps to discriminate confusing labels.

Table 12. Case Study for Confusing Event Labels

Model	Case 1: 出口额达到百万... The export volume reached millions.	Case 2: 双方通过对话解决分歧... The two sides resolved their differences through dialogue.
NPNs	(出口, Transfer-Money)	(对话, Meet)
TLNN	(出口, Transfer-Money)	(对话, Meet)
HGAT	(出口, Transfer-Money)	(对话, Meet)
L-HGAT	(出口, Transaction)	(对话, Contact)
Answer	(出口, Transaction)	(对话, Contact)

7 Conclusions

In this paper, we proposed a novel model, Label-Aware Heterogeneous Graph Attention Network (L-HGAT), for Chinese ED. Our model improves Chinese ED by addressing two problems, insufficient word-character interaction and event confusing. To alleviate the first problem, we built a heterogeneous word-character interactive graph, where characters and words are formulated as different types of nodes and connected with functional edges. Heterogeneous Graph Attention Network was then utilized to promote the information propagation between characters and words. To deal with the second problem, we designed a label-aware matcher, which introduces the semantic clues of event labels and employs a margin loss discriminating the confusing event labels. Experiments showed that L-HGAT could effectively solve the problems of word-character interaction and event confusing, and thus achieve 1.55% performance gain in F_1 over the competing approaches upon the ACE2005 and KBP2017 datasets on average. In the future, we would like to adapt L-HGAT into other tasks, like Chinese named entity recognition, to explore its performance.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Filatova E, Hatzivassiloglou V. Event-based extractive summarization. In *Proc. the 42nd Annual Meeting of the Association for Computational Linguistics Workshop on Summarization*, Jul. 2004, pp.104–111.

[2] Mitamura T, Liu Z Z, Hovy E H. Events detection, coreference and sequencing: What’s next? Overview of the TAC KBP 2017 event track. In *Proc. the 2017 Text Analysis Conference*, Nov. 2017.

[3] Ji H, Grishman R. Knowledge base population: Successful approaches and challenges. In *Proc. the 49th Annual Meeting of the Association for Computational Linguistics*, Jun. 2011, pp.1148–1158.

[4] Basile P, Caputo A, Semeraro G, Siciliani L. Extending an information retrieval system through time event extraction. In *Proc. the 8th International Workshop on In-*

*formation Filtering and Retrieval Co-Located with XIII AI*IA Symposium on Artificial Intelligence*, Dec. 2014, pp.36–47.

[5] Yang H, Chua T S, Wang S G, Koh C K. Structured use of external knowledge for event-based open domain question answering. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2003, pp.33–40. DOI: 10.1145/860435.860444.

[6] Chen Y B, Xu L H, Liu K, Zeng D J, Zhao J. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proc. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Jul. 2015, pp.167–176. DOI: 10.3115/v1/P15-1017.

[7] Nguyen T H, Grishman R. Event Detection and domain adaptation with convolutional neural networks. In *Proc. the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Jul. 2015, pp.365–371. DOI: 10.3115/v1/P15-2060.

[8] Nguyen T H, Grishman R. Modeling skip-grams for event detection with convolutional neural networks. In *Proc. the 2016 Conference on Empirical Methods in Natural Language Processing*, Nov. 2016, pp.886–891. DOI: 10.18653/v1/D16-1085.

[9] Liu S L, Chen Y B, Liu K, Zhao J. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*, Jul. 2017, pp.1789–1798. DOI: 10.18653/v1/P17-1164.

[10] Nguyen T, Grishman R. Graph convolutional networks with argument-aware pooling for event detection. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, *the 30th Innovative Applications of Artificial Intelligence (IAAI-18)*, *and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, Feb. 2018, pp.5900–5907. DOI: 10.1609/aaai.v32i1.12039.

[11] Liu X, Luo Z C, Huang H Y. Jointly multiple events extraction via attention-based graph information aggregation. In *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct. 2018, pp.1247–1256. DOI: 10.18653/v1/D18-1156.

[12] Chen Y B, Yang H, Liu K, Zhao J, Jia Y T. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct. 2018, pp.1267–1276. DOI: 10.18653/v1/D18-1158.

[13] Yan H R, Jin X L, Meng X B, Guo J F, Cheng X Q. Event detection with multi-order graph convolution and aggregated attention. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp.5766–5770. DOI: 10.18653/v1/D19-1582

[14] Cui S Y, Yu B W, Liu T W, Zhang Z Y, Wang X B, Shi J Q. Edge-enhanced graph convolution networks for event Detection with syntactic relation. In *Proc. the 2020 Findings of the Association for Computational Linguistics: EMNLP 2020*, Nov. 2020, pp.2329–2339. DOI: 10.18653/v1/2020.findings-emnlp.211.

[15] Lin H Y, Lu Y J, Han X P, Sun L. Nugget proposal networks for Chinese event detection. In *Proc. the 56th Annual Meeting of the Association for Computational Linguistics*, Jul. 2018, pp.1565–1574. DOI: 10.18653/v1/P18-1145.

[16] Ding N, Li Z R, Liu Z Y, Zheng H T, Lin Z B. Event detection with trigger-aware lattice neural network. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp.347–356. DOI: 10.18653/v1/D19-1033.

[17] Sui D, Chen Y B, Liu K, Zhao J, Liu S P. Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp.3830–3840. DOI: 10.18653/v1/D19-1396.

[18] Xue M G, Yu B W, Liu T W, Zhang Y, Meng E L, Wang B. Porous lattice transformer encoder for Chinese NER. In *Proc. the 28th International Conference on Computational Linguistics*, Dec. 2020, pp.3831–3841. DOI: 10.18653/v1/2020.coling-main.340.

[19] Xi X Y, Zhang T, Ye W, Zhang J L, Xie R, Zhang S K. A hybrid character representation for Chinese event detection. In *Proc. the 2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019. DOI: 10.1109/IJCNN.2019.8851786.

[20] Ahn D. The stages of event extraction. In *Proc. the 2006 Workshop on Annotating and Reasoning about Time and Events*, Jul. 2006.

[21] Ji H, Grishman R. Refining event extraction through cross-document inference. In *Proc. the 46th Annual Meeting of the Association for Computational Linguistics*, Jun. 2008, pp.254–262.

[22] Patwardhan S, Riloff E. A unified model of phrasal and sentential evidence for information extraction. In *Proc. the 2009 Conference on Empirical Methods in Natural Language Processing*, Aug. 2009, pp.151–160.

[23] Hong Y, Zhang J F, Ma B, Yao J M, Zhou G D, Zhu Q M. Using cross-entity inference to improve event extraction. In *Proc. the 49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, Jun. 2011, pp.1127–1136.

[24] McClosky D, Surdeanu M, Manning C. Event extraction as dependency parsing. In *Proc. the 49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, Jun. 2011, pp.1626–1635.

[25] Huang R H, Riloff E. Modeling textual cohesion for event extraction. In *Proc. the 26th AAAI Conference on Artifi-*

cial Intelligence, Jul. 2012, pp.1664–1670.

[26] Li P F, Zhu Q M, Zhou G D. Argument inference from relevant event mentions in Chinese argument extraction. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics*, Aug. 2013, pp.1477–1487.

[27] Li Q, Ji H, Huang L. Joint event extraction via structured prediction with global features. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics*, Aug. 2013, pp.73–82.

[28] Chen Z, Ji H. Language specific issue and feature exploration in Chinese event extraction. In *Proc. the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, *Companion Volume: Short Papers*, Jun. 2009, pp.209–212.

[29] Qin B, Zhao Y Y, Ding X, Liu T, Zhai G F. Event type recognition based on trigger expansion. *Tsinghua Science and Technology*, 2010, 15(3): 251–258. DOI: 10.1016/S1007-0214(10)70058-4.

[30] Li P F, Zhou G D, Zhu Q M, Hou L B. Employing compositional semantics and discourse consistency in Chinese event extraction. In *Proc. the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jul. 2012, pp.1006–1016.

[31] Li P F, Zhou G D. Employing morphological structures and sememes for Chinese event extraction. In *Proc. the 24th International Conference on Computational Linguistics*, Dec. 2012, pp.1619–1634.

[32] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In *Proc. the 5th International Conference on Learning Representations (ICLR)*, Apr. 2017.

[33] Zhang C X, Song D J, Huang C, Swami A, Chawla N V. Heterogeneous graph neural network. In *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019, pp.793–803. DOI: 10.1145/3292500.3330961.

[34] Wang X, Ji H Y, Shi C, Wang B, Ye Y F, Cui P, Yu P S. Heterogeneous graph attention network. In *Proc. the 2019 Web Conference*, May 2019, pp.2022–2032. DOI: 10.1145/3308558.3313562.

[35] Hu Z N, Dong Y X, Wang K S, Sun Y Z. Heterogeneous graph transformer. In *Proc. the 2020 Web Conference*, Apr. 2020, pp.2704–2710. DOI: 10.1145/3366423.3380027.

[36] Tu M, Wang G T, Huang J, Tang Y, He X D, Zhou B W. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, Jul. 2019, pp.2704–2713. DOI: 10.18653/v1/P19-1260.

[37] Hu L M, Yang T C, Shi C, Ji H Y, Li X L. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp.4821–4830. DOI: 10.18653/v1/D19-1488.

[38] Wang D Q, Liu P F, Zheng Y N, Qiu X P, Huang X J. Heterogeneous graph neural networks for extractive document summarization. In *Proc. the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp.6209–6219. DOI: 10.18653/v1/2020.acl-main.553.

[39] Jia R P, Cao Y N, Tang H Z, Fang F, Cao C, Wang S. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Nov. 2020, pp.3622–3631. DOI: 10.18653/v1/2020.emnlp-main.295.

[40] Wang G Y, Li C Y, Wang W L, Zhang Y Z, Shen D H, Zhang X Y, Henao R, Carin L. Joint embedding of words and labels for text classification. In *Proc. the 56th Annual Meeting of the Association for Computational Linguistics*, Jul. 2018, pp.2321–2331. DOI: 10.18653/v1/P18-1216.

[41] Zhang H L, Xiao L Q, Chen W Q, Wang Y K, Jin Y H. Multi-task label embedding for text classification. In *Proc. the 2018 Conference on Empirical Methods in Natural Language Processing*, Oct. 31–Nov. 4, 2018, pp.4545–4553. DOI: 10.18653/v1/D18-1484.

[42] Du C X, Chen Z Z, Feng F L, Zhu L, Gan T, Nie L Q. Explicit interaction model towards text classification. In *Proc. the 33rd AAAI Conference on Artificial Intelligence*, *the 31st Innovative Applications of Artificial Intelligence Conference*, *and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, Jul. 2019, pp.6359–6366. DOI: 10.1609/aaai.v33i01.33016359.

[43] Huang L F, Ji H, Cho K, Dagan I, Riedel S, Voss C. Zero-shot transfer learning for event extraction. In *Proc. the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2018, pp.2160–2170. DOI: 10.18653/v1/p18-1201.

[44] Lai V D, Nguyen T. Extending event detection to new types with learning from keywords. In *Proc. the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, Nov. 2019, pp.243–248. DOI: 10.18653/v1/d19-5532.

[45] Du X Y, Cardie C. Event extraction by answering (almost) natural questions. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Nov. 2020, pp.671–683. DOI: 10.18653/v1/2020.emnlp-main.49.

[46] Velickovic P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In *Proc. the 6th International Conference on Learning Representations*, Apr. 30–May 3, 2018.

[47] Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, 1967, 13(2): 260–269. DOI: 10.1109/tit.1967.1054010.

[48] Walker C, Strassel S, Medero J, Maeda K. ACE 2005 multilingual training corpus. Technical Report LDC2006 T06, Linguistic Data Consortium, 2006. https://catalog.ldc.upenn.edu/LDC2006T06, Jan. 2024 .

[49] Feng X C, Huang L F, Tang D Y, Ji H, Qin B, Liu T. A

language-independent neural network for event detection. In *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Aug. 2016, pp.66–71. DOI: 10.18653/v1/P16-2011.

[50] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2019, pp.4171–4186. DOI: 10.18653/v1/N19-1423.

[51] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *Proc. the 3rd International Conference on Learning Representations*, May 2015.

[52] Chen C, Ng V. Joint modeling for Chinese event extraction with rich linguistic features. In *Proc. the 24th International Conference on Computational Linguistics*, Dec. 2012, pp.529–544.

[53] Makarov P, Clematide S. UZH at TAC KBP 2017: Event nugget detection via joint learning with Softmax-margin objective. In *Proc. the 2017 Text Analysis Conference*, Nov. 2017.

[54] Zeng Y, Yang H H, Feng Y S, Wang Z, Zhao D Y. A convolution BiLSTM neural network model for Chinese event extraction. In *Proc. the 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and the 24th International Conference on Computer Processing of Oriental Languages, Natural Language Understanding and Intelligent Applications*, Dec. 2016, pp.275–287. DOI: 10.1007/978-3-319-50496-4_23.

Shi-Yao Cui received her B.E. degree in information security from Harbin Institute of Technology, Weihai, in 2018. She is currently a Ph.D. candidate of Institute of Information Engineering, Chinese Academy of Sciences, Beijing. Her research interests include event detection, event extraction, relation extraction and event relation identification.

Bo-Wen Yu received his Ph.D. degree in cyberspace security from Institute of Information Engineering, Chinese Academy of Sciences, Beijing, in 2022. He is currently an engineer in DAMO Academy, Alibaba Group, Beijing. His research interests mainly lie in information extraction, but also include metric learning and unsupervised learning.

Xin Cong received his B.E. degree in information security from University of Electronic Science and Technology of China, Chengdu, in 2018. He is currently a Ph.D. candidate of Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His research interests include few-shot learning, meta-learning and information extraction.

Ting-Wen Liu is a professor of the Institute of Information Engineering, Chinese Academy of Sciences, and the University of Chinese Academy of Sciences, Beijing. He is also a member of CCF. He received his Ph.D. degree in information security from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2013. His research interests include information extraction, text matching, knowledge graph, etc.

Qing-Feng Tan received his Ph.D. degree in information security from University of Chinese Academy of Sciences, Beijing, in 2017. He is an associate professor with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou. His current research interests include computer network and network security. Dr. Tan has published over 30 articles in conferences and journals. He is a member of CCF and IEEE.

Jin-Qiao Shi is a professor of the School of Cyber Security, Beijing University of Posts and Telecommunications, Beijing. He received his Ph.D. degree in computer architecture from Harbin Institute of Technology, Harbin, in 2017. His research interests include network measurement, intelligent information processing, big data security analysis, etc.