Hindawi
Scientific Programming
Volume 2021, Article ID 7466114, 13 pages
https://doi.org/10.1155/2021/7466114

Research Article

Software Knowledge Entity Relation Extraction with Entity-Aware and Syntactic Dependency Structure Information

Mingjing Tang,[1] **Tong Li**,[2] **Wei Wang**,[1] **Rui Zhu**,[1] **Zifei Ma**,[3] **and Yahui Tang**[4]

[1]*School of Software, Yunnan University, Kunming, China*
[2]*School of Big Data, Yunnan Agricultural University, Kunming, China*
[3]*School of Water Conservancy, Yunnan Agriculture University, Kunming, China*
[4]*School of Information, Yunnan University, Kunming, China*

Correspondence should be addressed to Tong Li; 3727@ynnu.edu.cn and Zifei Ma; zf_ma@hotmail.com

Received 8 September 2021; Revised 8 November 2021; Accepted 25 November 2021; Published 22 December 2021

Academic Editor: Francisco Ortin

Software knowledge community contains a large scale of software knowledge entities with complex structure and rich semantic relations. Semantic relation extraction of software knowledge entities is a critical task for software knowledge graph construction, which has an important impact on knowledge graph based tasks such as software document generation and software expert recommendation. Due to the problems of entity sparsity, relation ambiguity, and the lack of annotated dataset in user-generated content of software knowledge community, it is difficult to apply existing methods of relation extraction in the software knowledge domain. To address these issues, we propose a novel software knowledge entity relation extraction model which incorporates entity-aware information with syntactic dependency information. Bidirectional Gated Recurrent Unit (Bi-GRU) and Graph Convolutional Networks (GCN) are used to learn the features of contextual semantic representation and syntactic dependency representation, respectively. To obtain more syntactic dependency information, a weight graph convolutional network based on Newton's cooling law is constructed by calculating a weight adjacency matrix. Specifically, an entity-aware attention mechanism is proposed to integrate the entity information and syntactic dependency information to improve the prediction performance of the model. Experiments are conducted on a dataset which is constructed based on texts of the StackOverflow and show that the proposed model has better performance than the benchmark models.

1. Introduction

As a successful software knowledge community, StackOverflow provides a platform for software developers to exchange and share knowledge about software programming, configuration management, and project organization and gradually develops into an important knowledge base in the software field [1]. The social text of StackOverflow contains a large scale of specific software knowledge entities with complex structure and rich semantic relations. Extracting semantic relations of software knowledge entities and constructing knowledge graph for software engineering domain will promote the development of intelligent applications based on software knowledge graph, such as software document generation, software automatic question and answer, and software expert recommendation.

As a critical task of information extraction and knowledge graph construction, relation extraction refers to identifying the entity relations with uniform format from unstructured natural language text and associating the semantic relations with an entity pair, so as to promote the automatic construction of knowledge graph [2]. In the view of task decomposition, the relation extraction task can be divided into three subtasks: named entity recognition, trigger word recognition, and relation extraction [3]. Named entity recognition is to recognize and extract entities of specific meaning from text; trigger word recognition is to recognize and classify words that trigger entity relation; both of which are pretasks of relation extraction. In view of the development, relation extraction has experienced the early rule and dictionary-based method, machine-learning-based method, and deep-learning-based method [4], which

gradually developed from artificial feature engineering to automatic feature representations learning. Due to the avoidance of tedious manual feature extraction, relation extraction methods based on deep learning have become the current research focus, which can be divided into sequence-based method and dependency-based method. The methods based on sequence usually make use of Convolutional Neural Network (CNN) model or Recurrent Neural Networks (RNN) model to encode sentence sequences, which can enrich semantic representations of sentence sequences. The performance of relation extraction is improved, but the long-distance dependency of sentence sequences cannot be captured. The dependency-based method leverages GCN model to encode the syntactic features of sentence sequences, which can capture the global syntactic dependencies of sentence sequences. However, these models are inapplicable to specific domain with rich domain features, which lead to poorer performance of relation extraction.

At present, relation extraction has been widely used in financial investment, science education, biomedicine, and other fields, but relation extraction oriented to software engineering is still lacking. Due to the problems of entity sparse, relation ambiguity, and the lack of annotated dataset in the social text of software knowledge community, traditional relation extraction methods treat software knowledge entities and relations as regular textual content and fail to focus on the social and domain characteristics of software knowledge community text, resulting in the existing model not able to achieve the corresponding effect. It has become an important challenge in the field of software knowledge management to accurately and effectively extract software knowledge entities and semantic relations from the text of software knowledge community and construct a high-quality software knowledge graph.

Software knowledge entity relation extraction in this paper refers to the automatic identification of semantic relations among entities from unstructured software knowledge community text based on the software knowledge entity recognition, according to the predefined entity relation types. Therefore, the software knowledge entity relation extraction task based on such pipeline method can be modeled as a relation classification problem. For example, for sentence sequences in software knowledge community text with annotated entities, "*<e1>GetHashCode</e1> is method of base Object class of <e2> .Net Framework </e2>.*", the objective of the software knowledge entity relation extraction task is to predict the semantic relation "inclusion" between entity "*GetHashCode*" and entity "*.net Framework*" on the basis of a given entity pair ($e1$, $e2$), so as to obtain the relation triplet $<e1, R, e2>$.

Motivated by the above problems, we propose a novel hybrid model for software knowledge entity relation extraction with integrating entity-aware information and dependency structure information by combining the social characteristics and domain characteristics of the software knowledge community text. The main contributions of this paper are as follows:

(1) According to the structural characteristics of the social text in software knowledge community, Bi-GRU model is constructed to capture the context information of sentence sequences, and GCN model based on weight strategy is constructed to enhance the syntactic dependency representation of sentence sequences.

(2) To eliminate ambiguity resulting from polysemic words, combining entity information such as type, relative position, and entity mention, the entity-aware attention mechanism is proposed to integrate the entity information and syntactic dependency information of sentence sequence which improved the prediction performance of software knowledge entity relation classification.

(3) The annotated dataset for software knowledge domain covering 9532 entities, 17061 relation instances, and 5 predefined relation types is constructed based on texts of the StackOverflow.

(4) The proposed model achieves better performance than the benchmark models, which demonstrates its effectiveness.

The rest of this paper is organized as follows. In Section 2, we review related work. After that, the software knowledge entity relation extraction model is proposed in Section 3. The evaluation of the model and analysis of the obtained experimental results are included in Section 4. Finally, the main conclusions and future work are presented in Section 5.

2. Related Work

The early relation extraction method based on rules and dictionaries represents the semantic features of sentences by artificially constructing linguistic rules or domain dictionaries to realize entity recognition and discrimination of relation types [5, 6]. Such methods rely on domain knowledge and artificial feature extraction, and the performance of relation extraction depends on the quality and scale of rules or dictionaries, which has the characteristics of poor domain migration performance and low recall rate.

The machine learning-based relation extraction method utilizes feature engineering and annotated data to achieve better performance, which effectively alleviates the dependence on linguistics and domain knowledge, and has strong domain migration ability. According to whether annotated sample data are needed in the process of relation extraction, machine learning-based methods are divided into unsupervised, semisupervised, and supervised methods. Unsupervised machine learning method achieves automatic extraction of semantic relations by clustering semantically similar entities without manually predefined relation types and annotated sample data [7, 8]. Semisupervised machine learning methods utilize a small amount of annotated sample data to train large-scale unannotated corpus data and then extract semantic relations of entity pairs [9, 10]. Supervised machine learning methods regard relation

extraction as a classification problem and train classifiers to predict the semantic relation of entity pairs by manually annotating sample data, so as to realize relation extraction [11, 12].

Due to the avoidance of tedious manual feature extraction and improvement of error propagation in the feature extraction, relation extraction methods based on deep learning have become the current research focus. The relation extraction based on deep learning is mainly divided into sequence-based method and dependency-based method. Zeng et al. [13] used CNN model to extract lexical-level and sentence-level features, obtained sentence feature vector representations, and classified relations through the hidden layer and Softmax layer. Xu et al. [14] used the Long short-term memory (LSTM) network to integrate syntactic dependency tree, word vector feature, and part-of-speech feature with syntactic type feature and used the pooling layer and Softmax layer to classify the relation. To alleviate the problem of traditional neural network in syntactic dependency tree processing, Zhang et al. [15] introduced GCN model to model the syntactic dependency relation of sentence sequences and proposed a path-based pruning strategy to reduce data noise, which obtained a better performance on datasets SemEval-2010 Task 8 and TACRED.

In addition to the above general domain, relation extraction based on deep learning has also received extensive attention in vertical fields. In the biomedical field, Zhao et al. [16] obtained better performance in relation extraction by combining the n-ary relations extraction of gene-mutation drug, through applying GCN network and multihead attention mechanism to learn the dependency representations. In the software knowledge field, Zhu et al. [17] took tags of StackOverflow as a vocabulary and identified the inclusion relation between tags by extracting lexical features, word co-occurrence features, and topic features using semisupervised machine learning method. Zhao et al. [18] proposed a relation triplets extraction framework in the software engineering field by incorporating dependency parser with rule-based methods. In this framework, Support Vector Machine (SVM) is used as a classifier to evaluate the domain correlation of candidate relation triples, and a software knowledge graph covering 35,279 relation triples, 44,800 concepts, and 9660 verb phrases is constructed by combining text features, corpus features, concept features, and source features.

Above all, relation extraction and knowledge graph construction for software knowledge domain attracted more attention. However, compared with financial investment, science education, biomedicine, and other fields, corresponding publicly annotated dataset and proper models for software engineering field are not available.

3. The Proposed Method

In this section, we introduce a novel hybrid model for software knowledge entity relation extraction, named ED-SRE, which integrates entity-aware information and syntactic dependency structure information. ED-SRE model can be divided into input representation layer, context representation layer, syntactic dependency structure representation layer, feature fusion layer based on entity-aware attention mechanism, and linear classification layer. The overall architecture of the model is shown in Figure 1. The innovation of ED-SRE model is mainly reflected in the first, third, and fourth layers. As shown in Figure 1, the input sentence is fed into input representation layer to get multifeature representations as input for Bi-GRU model with the Bidirectional Encoder Representations from Transformers (BERT) model. Bi-GRU model captures contextual representation of sentence and as input for weighted GCN model. In syntactic dependency structure representation layer, a weighted adjacency matrix was constructed by calculating the weight coefficient of syntactic dependency between nodes. The feature fusion layer obtains final sentence representation through entity-aware attention and as input for linear classification layer.

3.1. Input Representation Based on Multifeature Fusion. Different from natural language processing tasks such as text classification and sentiment analysis, the task of relation extraction is more dependent on semantic features and entity information of sentence sequences. It is helpful to fully capture semantic features and entity information of sentence sequences in software knowledge community text to improve the performance of relation extraction. Semantic features of sentence sequence mainly include word vector features and part-of-speech (POS) tagging features, and entity information mainly includes entity type features and entity position features. According to relevant works [19, 20], ED-SRE model integrated semantic features of sentence sequences with entity information to enrich the input feature representations. The detailed description is as follows.

The feature representations of word vector are to transform the sentence sequence of software knowledge community text into low dimensional and dense distributed vector representations. Due to the significant social and domain characteristics of software knowledge community text, the problems of entity sparse and relation ambiguity are prominent. In order to capture the word vector representations of sentence sequences in different contexts and improve the performance of relation extraction, the BERT model was applied to pretrain the massive question-answer text of StackOverflow and take the pretrained word vector as the input of the model. In the training process of ED-SRE model, the word at time t in the input sequence can be represented by the corresponding word vector based on querying the BERT pretrained word vector, which is denoted as w_t^{emb}.

By using POS tagging tools, such as StanfordCoreNLP, the POS tagging feature of each word is obtained to analyze the sentence sequence of the software knowledge community text. Then, the POS tagging feature vector of the word at time t is denoted as w_t^{pos}.

In order to enhance the entity information representations of sentence sequences, the type of software knowledge entities is mapped to k-dimensional embedding vector, and

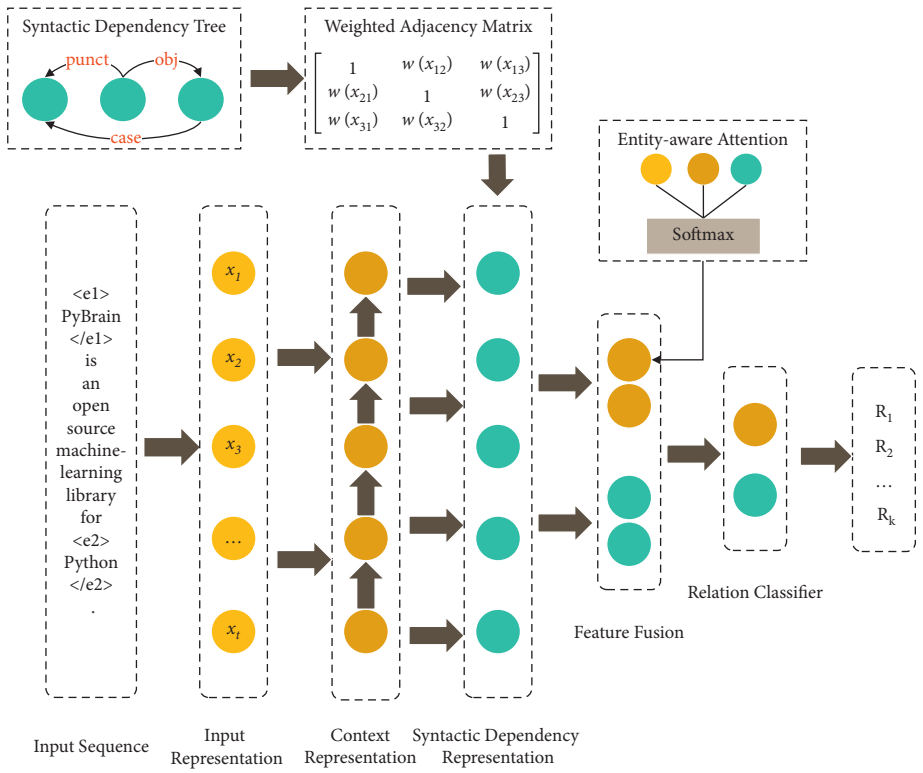

Figure 1: Overview of the proposed model.

then the type feature vector of entity pairs (e_s, e_o) is denoted as $[e_s^{\text{typ}}; e_o^{\text{typ}}]$. Entity position feature refers to the relative position information of each word and entity pair in the sentence sequence, which is mapped to the k-dimensional distance vector and then combined into the entity position feature of the current word, denoted as $[p_t^{e_s}; p_t^{e_o}]$. Therefore, the entity information feature vector of the word at time t of sentence sequence is denoted as $w_t^{\text{ent}} = [e_s^{\text{typ}}; e_o^{\text{typ}}; p_t^{e_s}; p_t^{e_o}]$.

Finally, by integrating the above word vector features, POS tagging features, and entity information features, the embedding vector of the input feature representation layer of the software knowledge entity relation extraction model is $\text{Int} = [w_t^{\text{emb}}; w_t^{\text{pos}}; w_t^{\text{ent}}]$.

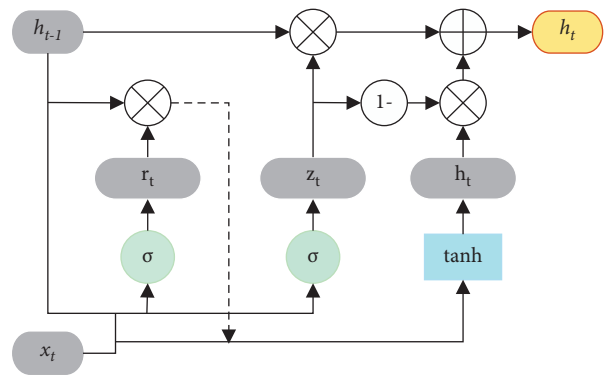

Figure 2: The structure of GRU circulation unit.

3.2. Context Representation Based on Bi-GRU. As the RNN model is very suitable for the modeling of contextual semantic features of sentence sequences, problems such as vanishing gradient and exploding gradient occur when processing long-distance sentence sequences. Therefore, several variant models of RNN have been proposed to alleviate these problems. By introducing the gating mechanism to control the information update, Cho et al. [21] proposed GRU model, which has the advantages of simple structure, fewer parameters, and fast training compared with the LSTM model. The circulation unit structure of GRU model is shown in Figure 2.

As shown in Figure 2, GRU model is streamlined on the basis of LSTM model, with only reset gate r_t and update gate Z_t. The reset gate r_t is used to control the degree to which the state information of the previous moment is discarded, where $r_t = 0$ shows that the hidden state information at the last moment is discarded; the update gate Z_t is used to control the degree to which the current state needs to retain the hidden state information of the previous moment, where $Z_t = 1$ shows that the hidden state information of the previous moment is retained. Given the sentence sequence $X = (x_1, x_2, \ldots, x_n)$ of software knowledge community text, the formal representations of GRU model at time t is shown in the following equation:

$$r_t = \sigma\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right), \tag{1}$$

$$z_t = \sigma\left(W_{xz}x_t + W_{hz}h_{t-1} + b_z\right), \tag{2}$$

$$\tilde{h}_t = \tan h\left(W_{xh}x_t + W_{hh}\left(r_t \odot h_{t-1}\right) + b_h\right), \tag{3}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \tag{4}$$

where σ and tan$h$ represent the nonlinear activation function of neurons, W represents the interlayer weight matrix, b is the bias vector of neurons, $\odot$ represents the matrix dot product, r_t represents the reset gate state, Z_t represents the update gate state, n is the number of cell units, $\tilde{h}_t$ represents the candidate hidden state vector, and h_t is the hidden state vector. According to the structure of GRU model, at the moment t of sequence processing, GRU model is able to capture the above information of the current sentence sequence through the states of reset gate and update gate. Nonetheless, it lacks the following information of the current sentence sequence, which also plays an important role in the task of software knowledge entity relation extraction. Therefore, two GRU models with opposite directions are used to construct a Bi-GRU model, which can simultaneously capture the context information of the sentence sequence at moment t. Assuming that the output of the forward GRU and reverse GRU at the current time t is $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively, the final output is obtained by combining the output of GRU model in both forward and backward directions, namely: $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.

3.3. Syntactic Dependency Structure Representation Based on GCN. As per the important structural information of sentence sequences, syntactic dependency structure information is widely used in the field of text mining. Obtaining the syntactic dependency structure information of sentence sequences is helpful to improve the performance and quality of relation extraction tasks [22, 23]. Since there are a large number of phrase entities and long-distance dependencies in software knowledge community text, it is not only necessary to capture the semantic features of sentence sequences, but also to mine the dependency structure information of sentence sequences in the process of relation extraction. Due to the defect of network structure, Bi-GRU model can effectively capture the context information of sentence sequence, but the ability to capture the long-distance dependent information of sentence sequence is limited. As a model for processing graph structural data, GCN model is proved to effectively capture long-distance dependent information by modeling graph nodes and edges and improve the effect of classification tasks [24]. Then, it has been applied to relation extraction tasks in the field of natural language processing [25, 26]. In this paper, GCN model is used to encode the syntactic dependency relationship of sentence sequence and treat each word in the sentence sequence as a node and the dependency relationship between words as an edge. The information of each node is updated by gathering the information of its neighboring nodes.

GCN model is formally defined as $G = (V, E)$, where V represents nodes set, namely all words in the sentence sequence; E represents edges set, that is the relationship between nodes. Given the sentence sequence $X = (x_1, x_2, \ldots, x_n)$ of software knowledge community text, the syntactic dependency tree of the sentence sequence can be generated through syntactic dependency analysis, and the adjacency matrix is used to represent the structural information. In this paper, the Stanford CoreNLP tool is adopted to analyze the syntactic dependency of sentence sequences in software knowledge community texts. For example, given the sentence sequence "*GetHashCode is method of base Object class of .Net Framework.*", the syntactic dependency tree of this sentence sequence is obtained through syntactic dependency analysis, as shown in Figure 3.

Syntactic dependency analysis is a structural analysis of sentence sequences at the syntactic level, which can accurately reflect the syntactic dependence between words in sentence sequences. According to the syntactic dependency tree, if there is a direct edge between nodes, it means that the distance between nodes is close and the syntactic dependence between words is strong. If there is a multihop indirect edge between nodes, it means that the distance between nodes is far, the syntactic dependence between words is weak, and it tends to decay as the distance increases. Existing studies have shown that extracting syntactic information from sentence sequences using pruning strategies leads to the loss of key information, resulting in performance degradation of downstream tasks. In order to represent the syntactic dependency structure information between words, we calculate the weight coefficient of syntactic dependency between nodes according to the distance between nodes and allocate weights to all edges of the syntactic dependency tree. Then, the weighted adjacency matrix A is constructed to transform the original syntactic dependency tree into the corresponding fully connected graph. The weighted adjacency matrix construction method is shown in Algorithm 1.

In the calculation process of the above algorithm, the input is the original syntactic dependency tree, the output is the weight adjacency matrix A, and i and j are the nodes of the syntactic dependency tree. When $i = j$, self-loop is added for each node, and then $A_{ij} = 1$. When $i \neq j$, the weight between nodes is calculated. The weight calculation process is as follows.

First, the distance between nodes is calculated; then, the weight coefficient of syntactic dependence between nodes is determined by function weight (d). In this paper, combined with the process that the syntactic dependence between nodes decreased exponentially with the increase of distance, Newton's Law of Cooling is used to calculate the correlation between the distance and the syntactic dependence of corresponding nodes, namely the weight. The function weight (d) is defined in the following equation:

$$\text{weight}(d) = \frac{\theta_{\text{init}}}{e^{\lambda(d-1)}}, \tag{5}$$

$$\lambda = \frac{1}{n} \ln \frac{\theta_{\text{init}}}{\theta_{\text{end}}}, \tag{6}$$

where θ_{init} represents the initial weight of the current node and is set to 1; θ_{end} indicates that the initial weight decays to the final weight with the increase of the distance between nodes, that is set as $(1/n)$, where n is the number of nodes.

Thus, for a GCN network with L-layer, the node i aggregates the features of the node itself and its neighboring nodes through graph convolution operation, and the output of feature vectors is shown in the following equation:

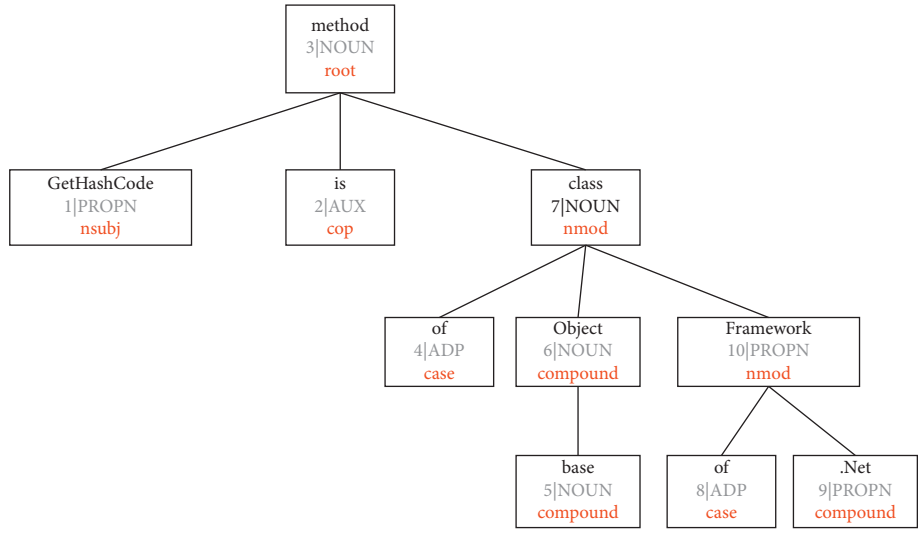

Figure 3: Example of a syntactic dependency tree.

$$h_i^l = \mathrm{ReLU}\left(\sum_{j=1}^{n} \frac{A_{ij} h_i^{l-1} W^l}{d_i + b^l}\right), \tag{7}$$

where ReLU represents the nonlinear activation function, h_i^{l-1} represents the input of the node at the l-th layer, h_i^l represents the output of the node at the l-th layer, W^l represents the weight matrix, $d_i = \sum_{j=1}^{n} A_{ij}$ is the degree of node i in the syntactic dependency graph, and b^l represents the bias units.

3.4. Feature Fusion Representation Based on Entity-Aware Attention. Due to the different objectives of natural language processing tasks such as text classification, machine translation, and information extraction, the selection of features is also different. For the relation extraction model, the context information and entity information of sentence sequence are the most critical features, which play an important role in improving the performance of relation extraction [27, 28]. Therefore, in the feature fusion layer of ED-SRE model, we propose an entity-aware attention mechanism to enhance the entity feature representations of sentence sequences. This attention mechanism makes full use of entity information such as entity type and relative position in sentence sequence and then improves the prediction performance of software knowledge entity relation extraction model.

The essence of attention mechanism is to selectively focus on some important information, which is a selection mechanism to allocate information processing capacity. Specifically, attention mechanism can be described as a mapping from a query in the target data to a series of key-value pairs in the metadata. By calculating the similarity between Query and Key, the corresponding weight of each value is obtained by the following equation [29]:

$$\mathrm{Attention}(Q, K, V) = \sum_{i=1}^{n} \mathrm{Sim}(Q_i, K_i) \cdot V_i, \tag{8}$$

where n is the length of the source data and Sim () is a function to evaluate the similarity, such as dot product calculation.

Then, the entity-aware attention mechanism is described as follows.

Firstly, given the sentence sequence $X = (x_1, x_2, \ldots, x_n)$ of software knowledge community text and the output of the sentence sequence after the L-layer GCN model is $H^l = (h_1^l, h_2^l, \ldots, h_n^l)$, then the weight of each word relative to the entity pair in the sentence sequence at the time t is obtained by the following equation:

$$a_t = \frac{\exp\left(h_t^l \cdot w_t^{\mathrm{ent}}\right)}{\sum_{j=1}^{n} \exp\left(h_t^l \cdot w_j^{\mathrm{ent}}\right)}, \tag{9}$$

where w_t^{ent} is the entity information feature vector representations of each word in the sentence sequence.

Then, the weighted sum is used to obtain the feature vector representations of the sentence sequence by the following equation:

$$H' = \sum_{i=1}^{n} a_i \cdot h_i. \tag{10}$$

In the feature fusion layer of ED-SRE model, in addition to the feature vector H' of sentence sequences, the feature vector of the head entity and tail entity of sentence sequences are integrated to obtain the feature fusion vector representations H_{fusion} of sentence sequences by the following equations:

$$h_s' = \eta(h_s), \tag{11}$$

$$h_o' = \eta(h_o), \tag{12}$$

$$H'' = \eta(H'), \tag{13}$$

$$H_{\mathrm{fusion}} = \left[H''; h_s'; h_o'\right], \tag{14}$$

where h_s represents the header entity feature vector representations, h_o represents the tail entity feature vector representations, and η represents the maximum pooling function.

3.5. Relation Classification. In the relation classification module of ED-SRE model, Feed Forward Neural Network (FFNN) is applied to obtain the final vector representations H_{sent} of sentence sequence, and the Softmax function is used to obtain the probability distribution of each relation type, so as to realize the classification of software knowledge entity relation. The calculation process is

$$H_{\text{sent}} = \text{FFNN}\left(H_{\text{fusion}}\right), \tag{15}$$

$$p\left(\frac{Y}{X}\right) = \text{softmax}\left(W_f H_{\text{sent}} + b_f\right), \tag{16}$$

where Y represents the set of predefined relational types, X represents sentence sequence, W_f represents weight matrix, and b_f represents bias units.

4. Experiment Results and Analysis

In order to evaluate the performance of ED-SRE model proposed in this paper, the ablation experiments and comparative experiments with the benchmark models in the field of relation extraction were carried out. ED-SRE model was implemented in Python using the deep learning framework PyTorch. It was specifically configured as Intel Xeon Gold 5117 processor, 2.0 GHz clock speed, NVIDIA Tesla T4 GPU, and 16GiB display memory, and all the experiments in this paper were conducted in this experimental environment.

4.1. Dataset Construction. Due to the lack of available annotated dataset for the software knowledge entity relation extraction task, we build annotated dataset required by the experiment based on the text of StackOverflow. In the StackOverflow, users label 1–5 tags according to the topic of the question, and the tags represent the knowledge domain of the question. Compared with Q&A text, tagWiki is a text with good text standardization and domain knowledge integrity, which used to describe the definitions of various tags and related resources in StackOverflow. Therefore, we construct the annotated dataset based on the Q&A text and tagWiki text of StackOverflow for software knowledge entity relation extraction. The detailed construction process is as follows.

Firstly, according to the popularity ranking of tags, 48 tags and their corresponding text content in tagWiki and Q&A text were randomly selected, and 19,000 sentences were obtained as corpus by natural language preprocessing technology. Combined with the classification of software knowledge entities, five predefined software knowledge entity relation types are obtained, including "use," "inclusion," "brother," "consensus," and "semantic." The detail information is shown in Table 1.

Then, the annotation tool of BRAT is used to annotate the corpus of software knowledge entity relation. The annotation team is composed of 10 teachers, software developers, graduate students, and undergraduates with software engineering background. After 5 rounds of cross-validation, the annotated dataset of software engineering domain is obtained. In order to ensure the scientific and reasonable results of model experiment, the dataset is divided into training set, verification set, and test set according to the ratio of 7 : 1 : 2 for the experiment of software knowledge entity relation extraction. The detailed information of the dataset is shown in Table 2.

4.2. Parameter Setting. In the training process of ED-SRE model, the dimension of pretrained word vector of the input representation layer is set as 360 dimensions, the number of units in the hidden state of Bi-GRU in the context coding layer is set as 200, and GCN is set as 1–3 layers. The Categorical Cross Entropy is used as the loss function of the model, and Adam is used as the optimizer. The initial learning rate is set at 0.5. At the same time, L2 regularization and Dropout mechanism were adopted to prevent overfitting of model training. The setting of relevant hyperparameters of the model is shown in Table 3.

4.3. Evaluation Metrics. General evaluation metrics in information extraction task are selected to evaluate the performance of the model, including precision rate (P), recall rate (R), and $F1$ score ($F1$). Precision rate (P) represents the percentage of correctly recognized samples in all recognized samples in the model recognition results; the recall rate (R) represents the percentage of correctly recognized samples in the number of all correct samples; $F1$ score is the weighted harmonic average of precision rate (P) and recall rate (R), which is used as the comprehensive performance evaluation index of the model. The formal expression of each evaluation index is as follows:

$$P = \frac{T_P}{T_P + F_P}, \tag{17}$$

$$R = \frac{T_P}{T_P + F_n}, \tag{18}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{19}$$

where T_P (True Positive) represents the number of correct relation types that are recognized as positive example by model, F_P (False Positive) represents the number of the wrong relation types that are recognized as positive example by model, and F_N (False Negative) represents the correct number of relation types that are recognized as the negative example by model.

4.4. Ablation Experiment and Analysis. In order to verify the effectiveness of the proposed ED-SRE model, which is based on the fusion of entity-aware information and dependency

Input: The syntactic dependency tree of sentence sequence, where V is nodes set and N is the number of nodes.
Output: Adjacency matrix A

```
(1)  Set A_ij = 0; initialize all elements of the adjacency matrix A to zero.
(2)  for each node i from V do:
(3)    for each node j from V do:
(4)      if j = i :
(5)        Set A_ij = 1;
(6)      else:
(7)      Calculate the distance d between node i and node j;
(8)      Set A_ij = Weight (d);
(9)    end for
(10) end for
```

Algorithm 1: Constructing the weighted adjacency matrix.

Table 1: The types of software knowledge entity relation.

Relation type	Meaning	Instance	Abbreviation
Use	Relation of using	Windows 10, Cortana	Use
Inclusion	Relation of including	Python, PyBrain	Inc
Brother	Relation of brother	C, Java	Bro
Consensus	Relation of synonym	JavaScript, JS	Con
Semantic	Other semantic relation	Azure Virtual Network, VPN	Sem

Table 2: The detail of the dataset.

	Training set	Verification set	Test set	Total
Use	2098	299	600	2997
Inclusion	3634	519	1038	5191
Brother	2318	331	662	3311
Consensus	2669	381	763	3813
Semantic	1224	175	350	1749

Table 3: Hyperparameters of the proposed model.

Names	Value
Word embedding dimension	360
Bi-GRU state size	200
GCN layer	1–3
Batch size	50
Num_epoch	200
Optimizer	Adam
Dropout	0.5
Learning rate	0.5

structure information, ablation experiments were performed on the software knowledge entity relation extraction annotated dataset. To ensure the fairness of the experimental results, Bi-GRU model was selected as the benchmark model during the ablation experiment, and the parameters of each model were consistent.

4.4.1. Contribution of Syntactic Dependency Structure to Model Performance. Based on the analysis of the syntactic dependency structure, we introduce GCN model to model the syntactic dependency structure information of sentence sequence and assign different weights to the adjacency matrix according to the distance between nodes, so as to realize the enhanced representations of syntactic dependency between nodes. Therefore, based on Bi-GRU model, we compare the performance of software knowledge entity relation extraction with GCN model and the weighted GCN model. The experimental results are shown in Table 4 and Figure 4.

Compared with Bi-GRU model, the F1 score of Bi-GRU-GCN model increases by 2.89%. It indicates that the introduction of the syntactic dependency structure information is helpful to improve the performance of the software knowledge entity relation extraction task. For Bi-GRU-GCN model, the F1 score of the model is increased by 2.5% after different weights are assigned to the edges between nodes, which indicates that the weight mechanism contributes to the performance improvement of the software knowledge entity relation extraction task. At the same time, the results show that when the layer number of GCN is set to 2, the precision rate and F1 score of the model are the highest, and the performance of the model is improved compared with 1-layer weight GCN model. However, when the number of GCN layers is set to 3, the precision rate and F1 score of the model decrease, indicating that the increase in the number of GCN layers will lead to overfitting.

4.4.2. Contribution of Entity-Aware Attention to Model Performance. In order to explore the contribution of entity-aware attention mechanism to the performance of software knowledge entity relation extraction task, ED-SRE model was selected to conduct comparative experiments with Bi-GRU, Bi-GRU-WGCN ($L = 2$), and other models. The experimental results are shown in Table 5 and Figure 5.

In Table 5, the model is labeled by symbol "✓," if the corresponding features representation is used. Otherwise, it is labeled "✕." Among them, word feature means whether the model uses BERT pretrained model for software

Table 4: Contribution of syntactic dependency structure to model performance.

Models	*P*%	*R*%	*F*1%
Bi-GRU	63.51	61.46	62.47
Bi-GRU-GCN	68.83	62.23	65.36
Bi-GRU-WGCN ($L = 1$)	70.35	65.54	67.86
Bi-GRU-WGCN ($L = 2$)	72.17	67.43	69.72
Bi-GRU-WGCN ($L = 3$)	71.14	66.72	68.86

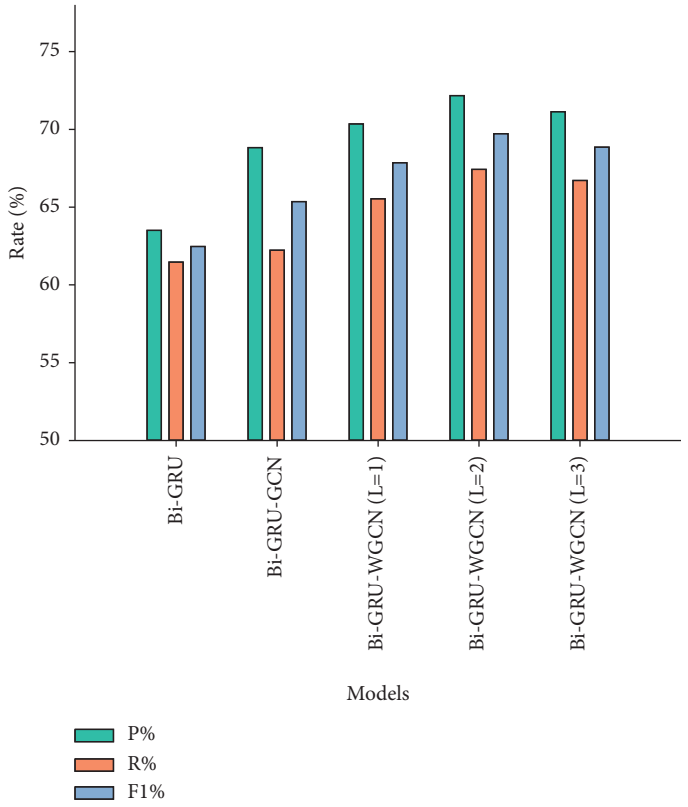

Figure 4: Comparison of experimental results of different models.

Table 5: Contribution of entity-aware attention to model performance.

Models	Word feature	Syntactic information	Entity information	*P*%	*R*%	*F*1%
Bi-GRU	×	×	×	63.51	61.46	62.47
Bi-GRU-WGCN ($L = 2$)	✓	✓	×	72.17	67.43	69.72
ED-SRE	✓	✓	✓	78.28	71.74	74.87

engineering domain. Based on the comparison experiments, the comprehensive performance of ED-SRE model is improved after the introduction of the entity-aware attention mechanism, and the F1 score is increased by 5.15% and 12.4%, respectively, compared with the other two models, which indicates that the semantic representations of sentence sequences can be effectively enhanced by extracting the entity information in sentence sequences and assigning different weights with the attention mechanism.

4.5. Comparative Experiment and Analysis. To evaluate the performance of ED-SRE model proposed in this paper, three classical relation extraction models were selected for comparative experiments from three aspects: sequence-based model, attention-based model, and syntactic dependency model. The experimental data are based on the software knowledge entity relation extraction annotated dataset constructed in this paper. Among them, Bi-LSTM-Position model [30] introduces the methods of maximum pooling and entity position-aware on the basis of Bidirectional Long Short-Term Memory (Bi-LSTM) model to extract general domain relations. Bi-LSTM-Attention model [31] only uses word vector features to extract general domain relations by introducing attention on the basis of Bi-LSTM layer. The CNN model [32] applies the convolutional neural network model to automatically learn the features of sentences through multiwindow filters for realizing the general

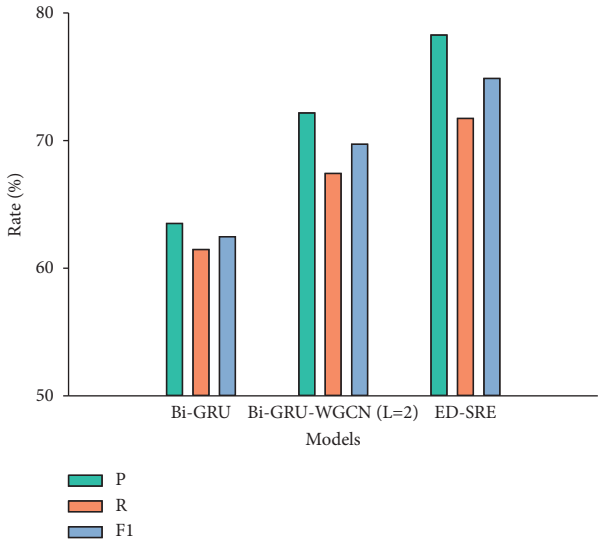

Figure 5: Comparison of experimental results with different models.

domain relation extraction task. The comparative experimental results are shown in Table 6 and Figure 6.

As shown in Table 6, the *F*1 scores of ED-SRE model are higher than all the other three comparison models. Compared with the Bi-LSTM-position model based on sequence, ED-SRE model can fully learn the syntactic features of sentence sequences by introducing the syntactic dependency structure information and effectively alleviate the long dependency problem of sentences. Compared with Bi-LSTM-Attention model based on the attention mechanism, ED-SRE model introduced the entity-aware attention mechanism, which can better represent the syntactic dependence between nodes and make full use of the influence of entity type, entity relative position, and other entity information on relation classification. Compared with CNN model based on syntactic dependency, ED-SRE model integrates the entity information and syntactic dependency structure information of sentence sequence, enriches the feature representations of sentence sequence, and is able to effectively improve the performance of software knowledge entity relation extraction task.

In addition, we compared the performance of the above models in each of the predefined relation types. The results are shown in Table 7 and Figure 7.

The performance of ED-SRE model is better than other comparison models in each relation type, and the highest *F*1 score is obtained in the inclusion relation type. At the same time, all models have good performance in inclusion, consensus, brother, and other relation types, but it has the lowest score in semantic relation type. The reason is that the semantic relation type represents other semantic relations, and there are relatively few relation instances in this relation type, which affects the accuracy of relation type prediction.

4.6. Construction and Application of Knowledge Graph. So far, based on the unstructured data and semistructured data of StackOverflow community text, this paper extracts software knowledge entity relation through the software knowledge entity relation extraction model ED-SRE and constructs a knowledge graph for the software knowledge domain. The overview diagram is shown in Figure 8.

Based on the software knowledge graph, it can be used in software engineering field such as entity-centric exploratory search, automatic software document generation, and software expert recommendation, which can help software

Table 6: Comparison of experimental results with different models.

Models	*P*%	*R*%	*F*1%
Bi-LSTM-Position	63.4	64.42	63.91
Bi-LSTM-Attention	63.08	65.21	64.13
CNN	62.96	64.47	63.71
ED-SRE	**78.28**	**71.74**	**74.87**

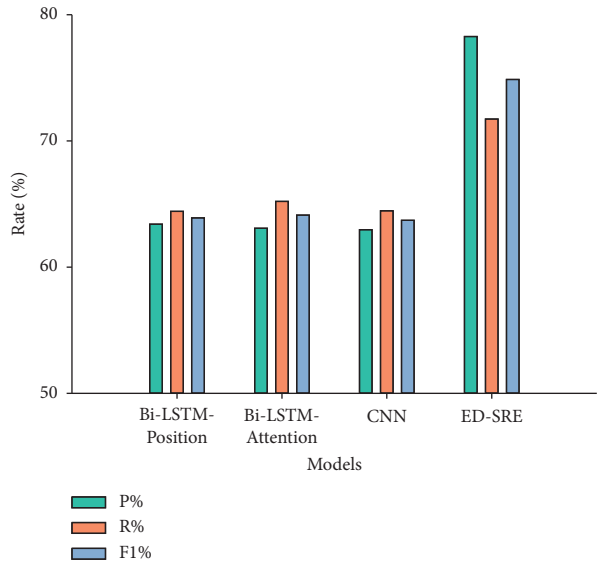

Figure 6: Comparison of experimental results with different models.

Table 7: Comparison of extracting effect of relation types.

Relation types	Bi-LSTM-Position *F*1 (%)	Bi-LSTM-Attention *F*1 (%)	CNN *F*1 (%)	ED-SRE *F*1 (%)
Use	65.73	62.08	61.36	69.23
Inclusion	67.54	70.46	69.25	84.35
Brother	62.75	62.83	63.73	75.58
Consensus	66.82	67.58	64.84	79.63
Semantic	56.69	57.72	59.36	65.57

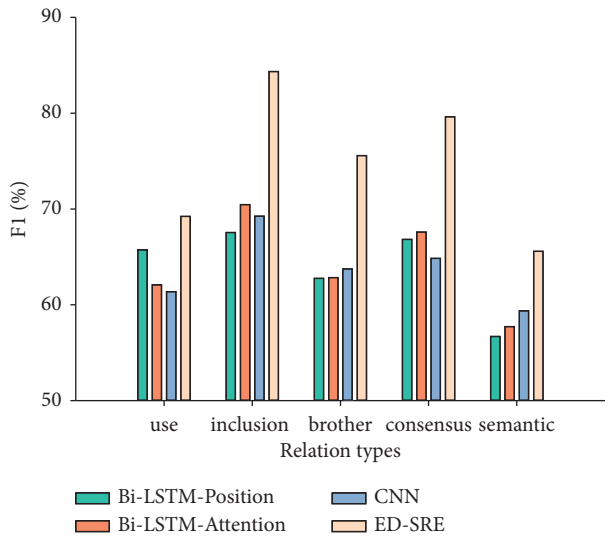

Figure 7: Comparison of extracting effect of relation types.

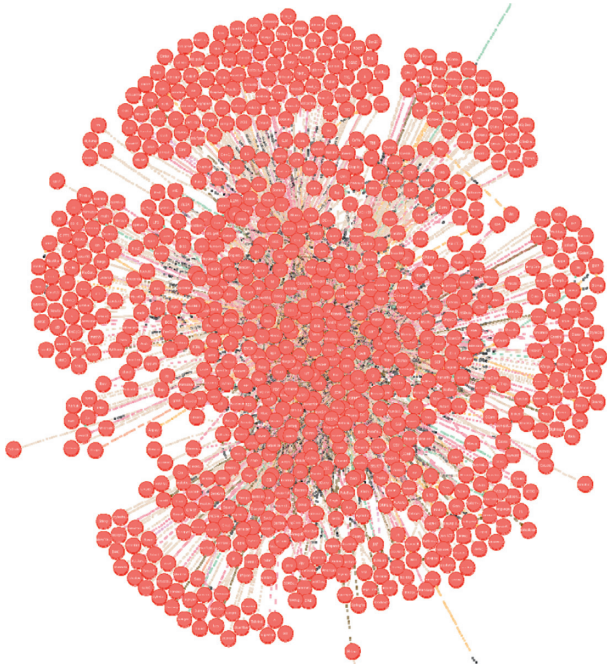

Figure 8: Software knowledge graph overview.

developers in solving the problems encountered in engineering practice. For example, the query language Cypher of Neo4j graph database is used to retrieve the entity "Python", and the software knowledge graph will graphically display

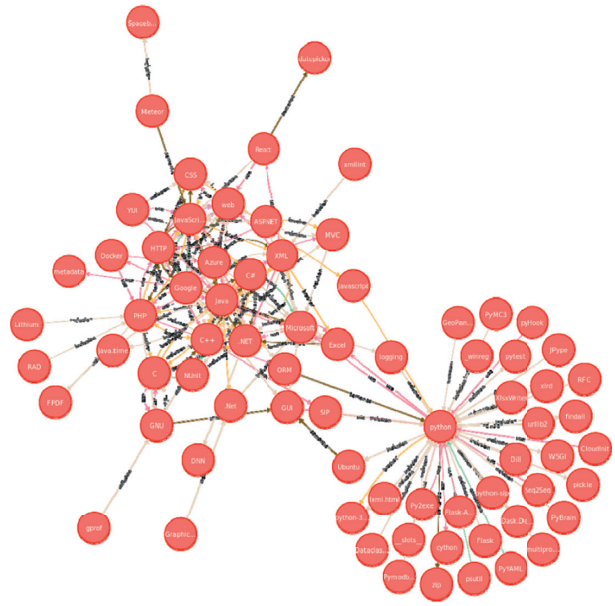

Figure 9: Search results and visualization of software knowledge graph.

various entities and relations related to "Python", as per the results shown in Figure 9.

5. Conclusion

In view of the problems of entity sparsity, relation ambiguity and the lack of annotated dataset in software knowledge domain, we proposed a novel software knowledge entity relation extraction model ED-SRE, which integrated entity-aware information and syntactic dependency structure information, to extract software knowledge entity relations from unstructured user-generated content. It utilized Bi-GRU model and GCN model to capture the contextual semantic representations and syntactic dependency representations of sentence sequences, respectively. To obtain more syntactic dependency information, a weight graph convolutional network based on Newton's cooling law is constructed by calculating the syntactic dependence between nodes. Combined with the information of entity type, relative entity position and entity mention, an entity-aware attention mechanism is proposed to integrate the entity information and syntactic dependency information of sentence sequence, so as to improve the prediction performance of software knowledge entity relation classification. To solve the problem of lack of open dataset, the annotated dataset for software knowledge domain covering

9532 entities, 17061 relation instances and 5 predefined relation types are constructed based on texts of the StackOverflow. The results of ablation experiments and model comparison experiments showed that ED-SRE model proposed in this paper has better performance than the benchmark model in the task of software knowledge entity relation extraction, and it paves a way for the application of software knowledge graph in the next step.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Mingjing Tang and Tong Li designed the study and wrote the paper. Zifei Ma and Rui Zhu performed the experiment. Wei Wang and Yahui Tang provided comments for revision. All authors reviewed and edited the manuscript, and all of them read and approved the final manuscript.

Acknowledgments

This work was supported by the Yunnan Science and Technology Major Project (grant no. 202002AE090010), and Subproject 5 of Yunnan Science and Technology Major (grant no. 202002AD080002-5).

References

[1] F. Wang, J. P. Liu, B. Liu, T. Y. Qian, Y. H. Xiao, and Z. Y. Peng, "Survey on construction of code knowledge graph and intelligent software development," *Journal of Software*, vol. 31, no. 1, pp. 47–66, 2020.

[2] D. M. Li, Y. Zhang, D. Y. Li, and D. Q. Lin, "Review of entity relation extraction methods," *Journal of Computer Research and Development*, vol. 57, no. 7, pp. 1424–1448, 2020.

[3] E. H. Hong, W. J. hang, S. Q. Xiao et al., "Survey of entity relationship extraction based on deep learning," *Journal of Software*, vol. 30, no. 6, pp. 1793–1818, 2019.

[4] X. Han, T. Y. Gao, Y. K. Lin et al., "More data, more relations, more context and more openness: a review and outlook for relation extraction," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 745–758, Suzhou, China, December2020.

[5] B. Deng, X. Z. Fan, and L. Q. Yang, "Entity relation extraction method using semantic pattern," *Computer Engineering*, vol. 33, no. 10, pp. 212–214, 2007.

[6] K. MarkóE, S. Schulz, O. Medelyan, and U. Hahn, "Bootstrapping dictionaries for cross-language information retrieval," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 528–535, Salvador, Brazil, August2005.

[7] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011, Suntec, Singapore, August2009.

[8] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 148–163, Barcelona Spain, September 2010.

[9] Z. Zhang, "Weakly-supervised relation classification for information extraction," in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pp. 581–588, Washington, DC, USA, November 2004.

[10] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 541–550, Portland, OR, USA, June 2011.

[11] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1083–1106, 2003.

[12] L. X. Gan, C. X. Wan, D. X. Liu, Q. Zhong, and T. J. Jiang, "Chinese named entity relation extraction based on syntactic and semantic features," *Journal of Computer Research and Development*, vol. 53, no. 2, pp. 284–302, 2016.

[13] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2335–2344, Dublin, Ireland, August 2014.

[14] Y. Xu, L. L. Mou, G. Li, Y. C. Chen, H. Peng, and Z. Jin, "Classifying relations via long Short term memory networks along shortest dependency paths," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1785–1794, Lisbon, Portugal, September 2015.

[15] Y. H. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2205–2215, Brussels, Belgium, October 2018.

[16] D. Zhao, J. Wang, H. F. Lin, Z. H. Yang, and Y. J. Zhang, "Biomedical cross-sentence relation extraction via multihead attention and graph convolutional networks," *Applied Soft Computing*, vol. 104, 2021.

[17] J. G. Zhu, B. J. Shen, X. Y. Cai, and H. F. Wang, "Building a large-scale software programming taxonomy from stackoverflow," in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, pp. 391–396, Pittsburgh, PA, USA, July 2015.

[18] X. J. Zhao, Z. C. Xing, M. S. Kabir, N. Sawada, J. Li, and S. W. Lin, "Hdskg: harvesting domain specific knowledge graph from content of webpages," in *Proceedings of the International Conference on Software Analysis, Evolution and Reengineering*, pp. 56–67, Klagenfurt, Austria, February 2017.

[19] Y. Chen, W. Yang, K. Wang, Y. Qin, R. Huang, and Q. Zheng, "A neuralized feature engineering method for entity relation extraction," *Neural Networks*, vol. 141, pp. 249–260, 2021.

[20] K. Deng and S. C. Wu, "Improving relation classification by incorporating dependency and semantic information," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, Glasgow, UK, July2020.

[21] K. Cho, B. Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[22] Z. He, W. Chen, Z. Li, W. Zhang, H. Shao, and M. Zhang, "Syntax-aware entity representations for neural relation extraction," *Artificial Intelligence*, vol. 275, pp. 602–617, 2019.

[23] C. Liu, X. T. Zhao, X. Li, and Y. W. Zhao, "Attention mechanism balances semantic representation and syntactic representation," in *Proceedings of the 9th International Conference on Computing and Pattern Recognition*, pp. 484–489, Xiamen, China, October 2020.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, November 2017.

[25] S. Vashishth, R. Joshi, S. S. Prayaga, C. Bhattacharyya, and P. Talukdar, "RESIDE: improving distantly-supervised neural relation extraction using side information," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1257–1266, Brussels, Belgium, October 2018.

[26] L. F. Song, Y. Zhang, Z. G. Wang, and D. Gildea, "N-ary relation extrac/tion using graph-state LSTM," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2226–2235, Brussels, Belgium, October 2018.

[27] H. Peng, T. Y. Gao, X. Han et al., "Learning from context or names? An empirical study on neural relation extraction," in *Proceedings of the Conference on Empirical Method AUs in Natural Language Processing (EMNLP)*, pp. 3661–3672, October 2020.

[28] L. Zhou, T. Y. Wang, H. Qu, L. Huang, and Y. G. Liu, "A weighted GCN with logical adjacency matrix for relation extraction," in *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, pp. 2314–2321, Santiago de Compostela, Spain, September 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31th Conference on Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, June 2017.

[30] D. X. Zhang and D. Wang, "Relation classification via recurrent neural network," 2015, https://arxiv.org/abs/1508.01006.

[31] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 207–212, Berlin, Germany, August 2016.

[32] T. H. Nguyen and R. Grishman, "Relation extraction: perspective from convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 39–48, Denver, CO, USA, June 2015.