applied sciences

MDPI

*Article*

# Entity-Centric Fully Connected GCN for Relation Classification

**Jun Long [1,2], Ye Wang [1], Xiangxiang Wei [1], Zhen Ding [1], Qianqian Qi [1], Fang Xie [1], Zheman Qian [1] and Wenti Huang [1,*]**

1 School of Computer Science and Engineering, Central South University, Changsha 410083, China; junlong@csu.edu.cn (J.L.); hai184612290@csu.edu.cn (Y.W.); xiangxaingwei@csu.edu.cn (X.W.); dingz9608@csu.edu.cn (Z.D.); qi1997@csu.edu.cn (Q.Q.); xiefang@csu.edu.cn (F.X.); 184612315@csu.edu.cn (Z.Q.)

2 Big Data Institute, Central South University, Changsha 410083, China

* Correspondence: huangwenti@csu.edu.cn

**Abstract:** Relation classification is an important task in the field of natural language processing, and it is one of the important steps in constructing a knowledge graph, which can greatly reduce the cost of constructing a knowledge graph. The Graph Convolutional Network (GCN) is an effective model for accurate relation classification, which models the dependency tree of textual instances to extract the semantic features of relation mentions. Previous GCN based methods treat each node equally. However, the contribution of different words to express a certain relation is different, especially the entity mentions in the sentence. In this paper, a novel GCN based relation classifier is propose, which treats the entity nodes as two global nodes in the dependency tree. These two global nodes directly connect with other nodes, which can aggregate information from the whole tree with only one convolutional layer. In this way, the method can not only simplify the complexity of the model, but also generate expressive relation representation. Experimental results on two widely used data sets, SemEval-2010 Task 8 and TACRED, show that our model outperforms all the compared baselines in this paper, which illustrates that the model can effectively utilize the dependencies between nodes and improve the performance of relation classification.

**Keywords:** graph convolutional network; relation classification; natural language processing

**Citation:** Long, J.; Wang, Y.; Wei, X.; Ding, Z.; Qi, Q.; Xie, F.; Qian, Z.; Huang, W. Entity-Centric Fully Connected GCN for Relation Classification. *Appl. Sci.* **2021**, *11*, 1377. https://doi.org/10.3390/app11041377

Academic Editor: Maxim Mozgovoy

Received: 24 December 2020

Accepted: 28 January 2021

Published: 3 February 2021

## 1. Introduction

Relation classification is an important task of natural language processing (NLP). It is the key step in many natural language applications. It extracts specific event or fact information from natural language text to help us automatically classify, extract and reconstruct massive content. This information usually includes entities, relations, and events. For example, extract time, location, and key person relations from news. Therefore, relation classification is widely used in information extraction [1,2], construction of knowledge base [3,4], and question answering [5,6] systems.

Most of the existing classification models are based on deep learning, such as recurrent neural networks(RNN), convolutional neural networks(CNN) and their improved models [7–12]. In the relation classification model, the natural language processing tools are employed to convert words in the text sequence into low-dimensional vectors. Then, the feature extractors employed to capture word representations and sentence-level semantic representations. Finally, the semantic relations between entities are predicated by using a classifier. Predicates are usually very important when predicating the relation between entities. If the distance between the entity and the predicate is too long, key information may be difficult to encode the semantic information. To solve this problem, a grammatical dependency tree is proposed to capture long-distance semantic information, simplify complex sentences and extract key information [13]. Xu et al. [14] applied long short term Memory networks (LSTM) on SDP between entities. Cai et al. [15] proposed a recurrent convolutional neural network that uses a two-channel LSTM to encode the global pattern in SDP, and uses CNN to capture the local features of every two adjacent words linked by dependencies.

To accurately obtain the semantic information between entities, early relation classification models used neural networks to obtain the shortest dependency path between entities. Yanxu et al. [14] proposed to apply LSTM to word sequences in the shortest path. Yangliu et al. [16] proposed to apply RNN to extract subtree features, and use CNN to extract shortest path features. Mito Makoto et al. [17] simplified the dependency tree to a subtree under the lowest common ancestor between entities. However, these models that run directly on the dependency tree are difficult to parallelize, so the computational efficiency is very low, because it is usually difficult to perform effective batch training on the tree.

Kipf et al. [18] presented the GCN, which can successfully process non-Euclidean data, and GCN model is also widely used in image recognition [19], visual question answering [20], and biology [21], and achieved good performance. However, the traditional GCN treats all nodes in the graph as equally important, ignoring the importance of entities in the sentence. And because GCN can only capture the information of direct neighbor nodes in one convolution process, GCN can only capture short-range context information. This problem can only be solved by adding a GCN layer. However, current research shows that increasing the number of GCN layers for relational classification tasks will make the model more complex [22]. Meanwhile, increasing the number of GCN layers will also cause excessive smoothness of node features, which will cause local features to converge to similar values.

Therefore, it is worth exploring and applying an extended GCN, which can capture the key information in the graph and eliminate some unnecessary noise. In order to solve the above GCN problem, we use the entity node as a global node, and propose an entity-centric fully connected GCN (FCGCN) model. The model in this paper uses an effective graph convolution operation to encode the dependency relation structure of the input sentence, and then extracts the entity-centric representation for reliable relation prediction.

We evaluate the model on the popular SemEval 2010 Task 8 dataset and TACRED dataset. Compared with previous models on these two data sets, the proposed model can make better use of the importance of entities in relation classification and obtain better results. When combined with the pruning strategy, the effect is further improved.

In short, the main contributions of this paper are as follows:

- We proposes a relation classification neural model based on graph convolutional network, using two entities as global nodes, that is, entity nodes have corresponding edges to other nodes.
- We use the difference vector of the entity pair as part of the relation classification constraint to make the relation classification result more accurate.
- A detailed analysis of the model and pruning technology shows that the pruning strategy and the proposed model have complementary advantages.

## 2. Materials and Methods

In this section, we propose a novel model for relation classification. The proposed model uses entity nodes as global nodes, and make full use of the importance of the entity itslef in relation classification to produce more accurate results.

The framework is mainly composed of four modules: (1) Sequence Encoding Module (2) Attention Module (3) Fully Connect Module (4) Relation Classification Module. The detail of our model is shown in Figure 1. The objective function of the proposed model consists of two parts, the loss of entity difference classifier $L_1$ and the loss of relation classifier $L_2$. $L_1$ is employed to train the difference vector of entity pair to be classified to the true relation type, and $L_2$ is employed to optimize the model by using the contextual semantic information of sentence.

### 2.1. Sequence Encoding Module

**Word embedding:** This paper uses GLOVE [23] to convert a sentence into a low-dimensional vector. Suppose $x = [x_1, x_2, ..., x_n]$ represents a sentence, where $x_i$ represents the i-th word and $n$ is the length of the sentence. The t-th word embedding in $s$ is denoted as $\boldsymbol{e}_i^w \in R^{d_w}$, $d_w$ is the dimension of the word embedding.

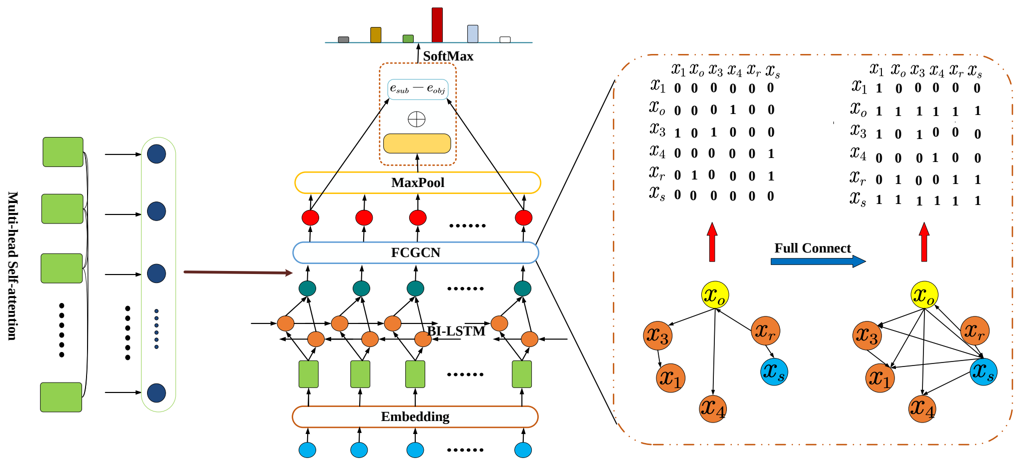

**Figure 1.** FCGCN model architecture diagram. The left part shows the overall architecture of FCGCN, and the right part shows FCGCN's entity-centric fully connected logical adjacency matrix, where $x_o$, $x_s$ represents the physical node and $x_r$ represents the root node.

**Position embedding:** Inspired by Zeng et al. [9], our work also considers the position feature of words.

There are two relative distances $e_i^{p_1} \in \mathbb{R}^{d_p}$ and $e_i^{p_2} \in \mathbb{R}^{d_p}$ between each word with entities $e_1$ and $e_2$ in the sentence. For example, $\langle e_1 \rangle$Jack Ma$\langle /e_1 \rangle$ is the founder of the Internet company $\langle e_2 \rangle$Alibaba$\langle /e_2 \rangle$, the relative distances of word company to "*Jack Ma*" and "*Alibaba*" are $-7$ and 1. We joint the word vector with the position vector to get the word vector representation. Therefore, the sentence can be expressed as follows

$$\boldsymbol{e}_i = \left[\boldsymbol{e}_i^w; \boldsymbol{e}_i^{p_1}; \boldsymbol{e}_i^{p_2}\right] \tag{1}$$

where $\boldsymbol{e}_i \in \mathbb{R}^m$, $m = d_w + 2d_p$, $d_w, d_p$ denotes the size of the vector.

**Bi-LSTM layer:** In order to encode the sequence features and context information of sentences, we connect the forward and reverse LSTM states to the BI-LSTM layer, The calculation formula is as follows

$$\overrightarrow{\boldsymbol{L}_i} = \overrightarrow{LSTM}(\boldsymbol{e}_i, \boldsymbol{L}_{i-1}) \in \mathbb{R}^{d_t} \tag{2}$$

$$\overleftarrow{\boldsymbol{L}_i} = \overleftarrow{LSTM}(\boldsymbol{e}_i, \boldsymbol{L}_{i+1}) \in \mathbb{R}^{d_t} \tag{3}$$

$$\overleftrightarrow{\boldsymbol{L}_i} = \left[\overrightarrow{\boldsymbol{L}_i} : \overleftarrow{\boldsymbol{L}_i}\right] \in \mathbb{R}^{2d_t} \tag{4}$$

where $d_t$ represents the dimension of the hidden layer of LSTM and $\overleftrightarrow{\boldsymbol{L}_i}$ represents a vector which containe the bidirectional semantic features.

### 2.2. Attention Module

The attention mechanism can automatically calculate the importance of different words in a sentence. Transformer model [24] shows that "self-attention" can achieve better results in sequence coding, it can compute the correlation between words in the sentence.

In a text sequence, the importance of words is often different. Especially entity nodes are important for relation classification. "Self-attention" can capture the internal structure of a sentence by learning the interaction between words in the sentence, and obtain the attention score between each word in the sentence. Therefore, more semantic information between words can be obtained through the "self-attention" mechanism. In this paper, we use multi-head attention to distinguish the importance of words in a sentence. The input given in this paper is the vector $\boldsymbol{s} = \left[\boldsymbol{e}_1^w, \boldsymbol{e}_2^w, \ldots, \boldsymbol{e}_n^w\right]$ after word embedding. In the self-attention process, we set query, key and value are equal to $s$, which is $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{s} \in \mathbb{R}^{n \times d_w}$. We get the hidden representation of the sentence as follows.

$$\boldsymbol{H} = Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = soft\max\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_w}}\right)\boldsymbol{V} \tag{5}$$

In the multi-head self-attention module, we pass Q, K, and V through a linear transformation, and input them into the scaled dot-roduct attention, and perform the above steps $h$ times. Therefore, the calculation of the i-th head is as follows

$$\boldsymbol{H}_i = Attention\left(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V\right)(i = 1, , , , , h) \tag{6}$$

where $\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V \in \mathbb{R}^{m \times d_m}$, $d_m = m/h$. The output of the multi-head attention layer is obtained by concatenating the outputs of $h$ heads, which is then input into a linear transformation to generate the final attention values.

$$\boldsymbol{M} = MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{W}^o Concat(\boldsymbol{H}_i, , , , , \boldsymbol{H}_h) \tag{7}$$

where $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n]$ is the hidden representations sentences generated by multi-head self-attention, $\boldsymbol{W}^o \in \mathbb{R}^{d_t \times d_t}$ represents the weight matrix.

To make better use of the information captured by multi-head self-attention, the same feedforward network is employed to the $\boldsymbol{m_i}$ in the multi-head self-attention output $\boldsymbol{M}$. The calculation method is as follows

$$FFNN(\boldsymbol{m}_i) = p(\boldsymbol{m}_i \boldsymbol{M}_1 + \boldsymbol{\beta}_1)\boldsymbol{M}_2 + \boldsymbol{\beta}_2 \tag{8}$$

where $\boldsymbol{M}_1 \in \mathbb{R}^{d \times d_q}$, $\boldsymbol{M}_2 \in \mathbb{R}^{d_q \times d}$ represents the conversion matrix. $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ represents the bias, and $d_q$ represents the size of the hidden layer. Inspired by Vaswani et al. [24], we integrate the multi-head self-attention and the output of the feedforward layer through the residual connection [25], and then perform layer normalization [26]. The calculation method is as follows

$$\boldsymbol{C} = LayerNorm\left(\acute{\boldsymbol{M}} + FFNN\left(\acute{\boldsymbol{M}}\right)\right) \tag{9}$$

where $\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\} \in \mathbb{R}^{n \times d}$, $\acute{\boldsymbol{M}} = LayerNorm(\boldsymbol{M} + \boldsymbol{s})$ represents the residual connection between sentence embedding and multi-head self-attention layer output.

We obtain the final word attention results as follows

$$\alpha_t = \frac{\exp(\boldsymbol{v}_t)}{\sum_{j=1}^n \exp(\boldsymbol{v}_j)} \tag{10}$$

$$\boldsymbol{v}_i = \boldsymbol{h}_i \boldsymbol{W} \boldsymbol{r} \tag{11}$$

where $\boldsymbol{W} \in \mathbb{R}^{d_w \times d_w}$ represents square matrix, and $\boldsymbol{r} \in \mathbb{R}^{d_w \times 1}$ represents a random query vector.

### 2.3. *Fully Connect Moudle*

The graph convolutional network [18] is an adaptation of the convolutional neural network [27] for encoding graphs. We can employ $n \times n$ adjacency matrix A to represent a graph with $n$ nodes. If there is an edge from node $i$ to node $j$ which means $A_{ij} = 1$. For the convolution operation of node $i$ in the $l$-$th$ layer, we use the $l$-$1th$ layer node to represent the input vector $\boldsymbol{h}_i^{(l-1)}$, and use $\boldsymbol{h}_i^{(l)}$ to represent the output vector. The convolution formula is as follows

$$\boldsymbol{h}_i^{(l)} = \rho\left(\sum_{j=1}^{n} A_{ij} W^{(l)} \boldsymbol{h}_j^{(l-1)} + \boldsymbol{b}^{(l)}\right) \tag{12}$$

where $\boldsymbol{W}^{(l)}$ represents the weight matrix, $\boldsymbol{b}^{(l)}$ represents the bais vector, and $p$ represents the activation function (e.g., RELU).

To emphasize the importance of entities in the dependency tree. Inspired by Guo et al. [28], this paper proposed a new extended dependency tree, in which entities are used as global nodes, and the global nodes directly connect with other nodes. Since the node cannot connect to itself in the dependency relation, the information of $\boldsymbol{h}_i^{(l-1)}$will never be transferred to $\boldsymbol{h}_i^{(l)}$. It is necessary to add self-citation to itself. The schematic diagram is shown in Figure 2.

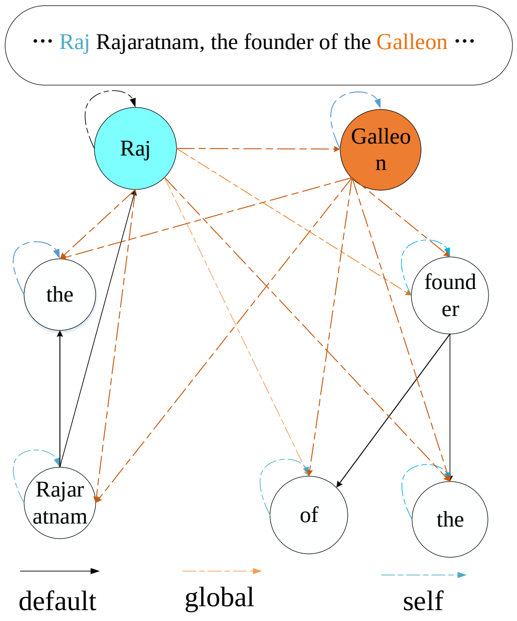

**Figure 2.** Fully connected connection graph.default represents the default connection in the grammatical dependency tree, global represents the entity node and other nodes have corresponding connections, self represents self-citing.

The global nodes have three advantages. First, it emphasizes the importance of entity nodes in relation classification, and employ entity information to improve the accuracy of relation classification. Secondly, global nodes provide each node with a global view of the input graph, so that the global node can not only gather the information of its neighbor nodes, but also gather the information of all other nodes in a convolution operation. Finally, global nodes can help other nodes gather information, which can promote the process of node information dissemination. We combine the pruning strategy proposed by Zhang et al. [29], which can allow entity nodes to gather more key information in sentences and eliminate irrelevant information. Therefore, we modify the convolution calculation to

$$\boldsymbol{h}_i^{(l)} = \rho\left(\sum_{j=1}^{n} \boldsymbol{\alpha}_j \left(\tilde{\boldsymbol{A}}_{ij} W^{(l)} \boldsymbol{h}_j^{(l-1)} / d_i + \boldsymbol{b}^{(l)}\right)\right) \tag{13}$$

$$\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I} \tag{14}$$

where $\boldsymbol{h}_1^{(0)}, \boldsymbol{h}_2^{(0)}, \ldots, \boldsymbol{h}_n^{(0)} = \boldsymbol{s} = \left[\boldsymbol{e}_1^w, \boldsymbol{e}_2^w, \ldots, \boldsymbol{e}_n^w\right]$, $d_i = \sum_{j=1}^{n} A_{ij}$ represents the out-degree of the node $i$, and $I$ represent the matrix of $n \times n$ global and other nodes with edges.

### *2.4. Relation Classification Module*

After passing through the L-level GCN, we get the hidden representation of each node. In order to use the hidden representation of these nodes to represent the relation classification, we first obtain the following sentence representation

$$\boldsymbol{h}_{sent} = \max pool\left[H^{(L)}\right] \tag{15}$$

where $H^{(L)} = \left[\boldsymbol{h}_1^{(l)}, \boldsymbol{h}_2^{(l)}, ..., \boldsymbol{h}_n^{(l)}\right]$ represents the matrix representation of the sentence after the L-layer FCGCN.

Lin et al. [30] regard the relation $\boldsymbol{r}$ as the transformation from the head entity to the tail entity ($\boldsymbol{e}_{sub} + \boldsymbol{r} \approx \boldsymbol{e}_{obj}$) and use the margin-based ranking criterion to learn their embeddings (i.e., entities and relations). Finally, their model simply used the corresponding embedding representation vector calculation to predict the relation between two target entities, and achieved good results. These studies have fully evaluated that the difference in word embedding of entity pairs can reflect the semantic information of the relation between them. Therefore, we believe that the difference vector of entities can also provide effective evidence for relation classification. Specifically, given a sentence marked with $\boldsymbol{r} = \boldsymbol{e}_{sub} - \boldsymbol{e}_{obj}$, we use the difference vector of the entity pair as part of the relation classification constraint and get the predicted probability of the entity as follows

$$P(y|\boldsymbol{r}) = soft\max(MLP(\boldsymbol{r})) \tag{16}$$

where $\boldsymbol{e}_{sub}$ and $\boldsymbol{e}_{obj}$ represent the vector of the entity obtained after FCGCN, $MLP(.)$ represents the multilayer perceptron. Finally, the vector obtained after the entity phase is cut and $h_{sent}$ are joined to obtain the final vector. The formula is as follows

$$\boldsymbol{h}_{out} = [\boldsymbol{h}_{sent} : \boldsymbol{r}] \tag{17}$$

We feed the hidden representation of sentence information into the feedforward neural network through the softmax operation to obtain the probability distribution of the following relationship

$$\boldsymbol{h}_{final} = FFNN(\boldsymbol{h}_{out}) \tag{18}$$

$$P\left(y|\boldsymbol{h}_{final}\right) = soft\max\left(\boldsymbol{h}_{final}\right) \tag{19}$$

### *2.5. Module Training*

In this framework, the objective function of the proposed model FCGCN includes two parts, the loss of entity classifier $L1$ and the loss of relation classifier $L2$. Suppose there are N pieces of data in the training set $T = \{T_1, T_2, ..., T_N\}$ and their corresponding labels $\{y_1, y_2, ..., y_N\}$, We use cross entropy to solve the loss function of the difference vector classifier $L1$

$$L_1 = -\sum_{i=1}^{N} \log P(y_i|\boldsymbol{r}) \tag{20}$$

Similarly, we also use the cross entropy shown below to get the loss of the relationship classifier $L2$

$$L_2 = -\sum_{i=1}^{N} \log P\left(y_i|\boldsymbol{h}_{final}\right) \tag{21}$$

Finally, we use the l2-norm to constrain the minimized loss function $L$

$$\min\ L = L_1 + L_2 \tag{22}$$

In this paper, the stochastic gradient descent method is used to optimize.

## 3. EXPERIMENT

### *3.1. DateSets*

We verify the performance of FCGCN on two widely used relation classification datasets: TACRED and SemEval 2010 Task 8.

**TACRED:** The TACRED dataset contains 106,264 instances and 42 relation types (including 41 predefined semantic relations and a "None" relation, the "None" relation means that there is no any relationship between an entity pair) [31]. Each instance of the dataset is a sentence that contains a pair of entity mentions, and it is labeled by one of the 42 relation types.

**SemEval 2010 Task 8:** The SemEval 2010 Task 8 is very popular recently in dealing with the problem of relation classification, which contains 10,717 instances and 9 relation types and a particular "other" relation type [32]. This dataset is very small, which is only 1/10 of TACRED. Each instance in this dataset expresses a sematic relation between two marked entities. 8000 instances are contained in the training set and 2717 instances are contained in the test set.

Based on these two datasets, we employ pre-trained 300-dimensional Glove [23] to initialize the word embeddings. The pruning path K is set to 1, and the batch size is 50. The model is trained for 100 epochs. The details of the selected hyperparameters are summary in Table 1.

**Table 1.** Hyper-parameters used in the experiments.

| Para | Description | SemEval | TACRED |
|---|---|---|---|
| $d_w$ | *Word embedding* | 300 | 300 |
| $d_p$ | *Position embedding* | 30 | 30 |
| $\gamma$ | *Initial learning rate* | 1.0 | 0.5 |
| $\zeta$ | *decay rate* | 0.9 | 0.95 |
| $\xi$ | *Initial dropout rate* | 0.5 | 0.5 |
| $h_l$ | *Bi-LSTM hidden size* | 100 | 100 |
| $h_f$ | *FCGCN hidden size* | 200 | 200 |
| $d_l$ | *FCGCN layers* | 2 | 2 |
| $h$ | *Attention heads* | 3 | 3 |

### *3.2. Performance Comparison*

The performances of the proposed model and baselines on TACRED dataset are illustrated in Table 2. We employ four types of models as the baselines: (1) Logistic regression (LR) based model is a traditional method for relation classification, which utilizes the dependency information and the lexical information of sequences to form sentence-level features. (2) The convolutional neural network based model proposed by Nguyen et al. [10] is employed to deal with the relation classification task, which develop a feature extractor to automatically capture the semantic features of sentences for relation classification. (3) The long short term memory (LSTM) based methods, including PA-LSTM [31] model, tree-LSTM model [33] and SDP-LSTM model [14], are introduced as the baselines. The PA-LSTM model employs a position-aware attention mechanism and a LSTM encoder to extract the sentence features.

The SDP-LSTM model focuses on the shortest dependency path between an entity pair, in which the key semantic information is contained. The tree-LSTM model encodes the whole tree structure to obtain the semantic information of sentences. (4) The graph convolutional network based models C-GCN proposed by Zhang et al. [29] and AGGCN proposed by Zhang et al. [34] are also introduced as a baseline in our experiment.

From Table 2, we can observe that the GCN based models achieve better performances that other baselines and our propose method outperforms all the baselines. Our model achieves significant improvement in terms of F1 with at least 0.7. The CNN based model has the best precision score 75.6 and the lowest recall score with only 47.5. Compared to AGGCN and C-GCN, the proposed FCGCN achieves better performance. We believe the reason behind it is that the global entity nodes can effectively accumulate sematic information from the whole dependency tree and form more informative semantic features. The experimental results prove the effectiveness of the proposed FCGCN model.

In order to evaluate the universality of the proposed proposed model, we also conducted experiments on SemEval 2010 Task 8. We mainly carried out experiments on some dependency models, as shown in Table 3. The proposed model can still get the F1 score 86.0 in SemEval 2010 Task 8, and and outperforms the compared baselines.

**Table 2.** Results on TACRED.

| **Model** | **P** | **R** | **F1** |
|---|---|---|---|
| LR | *73.5* | 49.9 | 59.4 |
| CNN | ***75.6*** | 47.5 | 58.3 |
| SDP-LSTM | *66.3* | 52.7 | 58.7 |
| Tree-LSTM | *66.0* | 59.2 | 62.4 |
| PA-LSTM | *65.7* | **64.5** | 65.1 |
| C-GCN | *69.9* | 63.3 | 66.4 |
| AGGCN | *69.9* | 60.9 | 65.1 |
| FCGCN (ours) | *72.2* | 62.0 | **67.1** |

**Table 3.** Results on SemEval 2010 Task 8.

| **Model** | **F1** |
|---|---|
| CNN (Zeng et al., 2014) | *83.7* |
| SDP-LSTM (Xu et al., 2015b) | *84.4* |
| Tree-LSTM (Tai et al., 2015) | *82.7* |
| LR (Zhang et al., 2017) | *82.2* |
| PA-LSTM (Zhang et al., 2017) | *84.8* |
| C-GCN (Zhang et al., 2018) | *84.8* |
| C-AGGCN (Zhang et al., 2020) | *85.7* |
| FCGCN (ours) | ***86.0*** |

## *3.3. Ablation Study*

To demonstrate the effectiveness of various components of the propose model, we implement a series of ablation experiments on the TACRED dataset. The results of ablation study are shown in Table 4, from which we can observe that the performance of the proposed model drops with removing the components. Especially removing the pruning strategy, the F1 score decreases 2.2, which indicates the importance of the pruning mechanism. The performance drops by 0.7 and 0.9 respectively when removing the Bi-LSTM layer and multi-head attention layer. It is worth noting that the performance drops by 1.5 when masking the entities with random tokens, which indicates that the semantic information contained in the entity mentions is crucial for relation classification.

**Table 4.** Ablation study on TACRED.

| **Model** | **Dev F1** |
|---|---|
| FCGCN | ***67.1*** |
| -Pruning category | *64.9* |
| -Bi-LSTM layer | *66.4* |
| -Multi-head layer | *66.2* |
| -Mask-entity | *65.6* |

**Effect of Pruning:** Figure 3 shows the effect of the FCGCN model with pruning tree, where K indicates that the distance between the token contained in the pruning tree and the dependent path in the LCA subtree is at most K. In order to show the path-centric pruning effect, when the pruning distance K changes, we compare the experimental results under different epochs. We conducted experiments on $K \in \{0, 1, 2, \infty\}$ on the TACRED development set, and compared the models without pruning. As shown in Figure 3, when K = 1, the performance of all epochs reaches the peak, which also evaluates that pruning improves the proposed model.

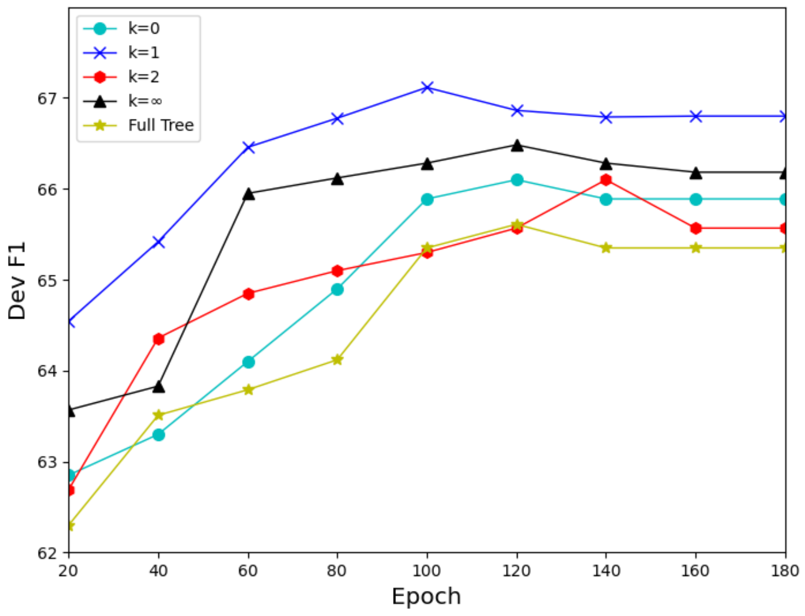

**Figure 3.** Experimental results in terms of F1 under different epochs whih different pruning strategies.

**Effect of Mask-entity:** Figure 4 shows the effect of the FCGCN model with and without concealed entities.To show the role of the entity itself in the FCGCN model, we compared the experimental results under different epochs. As shown in Figure 4, the entity is not covered, and the performance of all epochs reaches the peak, which also evaluates the importance of the entity in relation extraction.

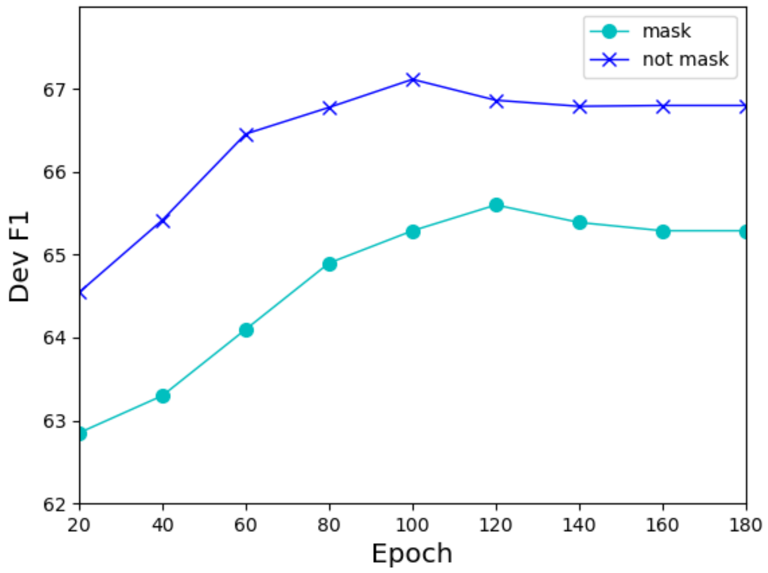

**Figure 4.** Experimental results in terms of F1 under different epochs with and without masking the entities.

**Analyze of Bi-LSTM & Multi-Head Attention:** In the relation classification model based on deep learning, Bi-LSTM and multi-head attention mechanism are widely used in the model. By capturing the forward and backward LSTM information, each word in the sentence can capture contextual semantic information. Multi-head attention mechanism can help us pay more attention to the important words in the sentence, rather than some words that can bring noise. This model uses multiple queries to calculate in parallel to select multiple pieces of information from the input information to assign attention weights to each word in the sentence, playing the role of global observation. As shown in Table 4, the F1 scores of Bi-LSTM and Multi-Head Attention in the proposed model indicate that BI-LSTM has complementary effects in capturing sequence information, multi-head attention mechanism in global correlation and acquiring more semantics between words. Combining Bi-LSTM and multi-head attention mechanism can obtain richer word-level and sentence-level representations, capture more semantic information, and can improve the accuracy of relation classification.

### *3.4. Effect of Hyper-Parameters*

In this section, the influence of two hyperparameters in the method proposed in this paper is discussed through experiments, the number of attention heads $h$ and the number of graph convolutional layers $d_l$.

Since multi-head attention allows the model to collectively pay attention to information from different representation subspaces from different locations. Choosing the right number of attention heads is crucial to the proposed model. As can be seen from the Figure 5a, as the number of attention heads increases, the performance improves. Using 3 heads in the multi-head self-attention layer produces the best performance. Use more than 3 heads, each additional head, performance will gradually decrease.

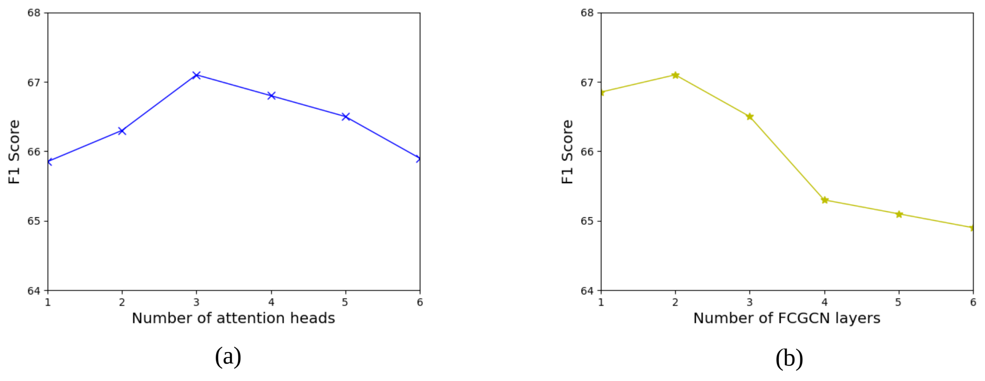

**Figure 5.** Effect of hyper-parameters. (**a**) Results on different number of attention heads $h$. (**b**) Results on different Number of FCGCN layers $d_l$.

Then, we also study the influence of the number of graph convolution layers when the number of attention heads is fixed at 3. It can be seen from the Figure 5b that as the number of GCN layers increases, the performance improves. The best effect is achieved when using two layers in FCGCN. Using more than 2 layers, each additional layer, and the multi-layer GCN algorithm used for relation classification tasks will produce high space complexity.

## 4. Conclusions

In this paper, we introduced a novel entity-centric fully connected graph convolutional network (FCGCN) for relation classification. In FCGCN, we made global nodes directly connect with other nodes. This operation emphasizes the importance of entity nodes for relation classification, especially when the entities not be masked, the performance of the model is greatly improved. In this way, the information of all nodes in the graph can be directly obtained by performing only 1 layer of FCGCN. Moreover, the difference vector of the entity pair was introduced to constraint the relation between entity pair, which can effectively improve the relation classification results. The experimental results on the TACRED and the SemEval 2010 task 8 datasets show that FCGCN use the semantic information in the dependency tree more comprehensively, and outperformed better results than baselines. We also found that the F1 value reached 67.1 when combined with the dependency tree after using the pruning strategy, which shows that the combination of them further improves the model.

In future work, we will explore a new pruning strategy to combine with the proposed model to reduce noise as much as possible, thereby further improving the performance of the model.

**Author Contributions:** Conceptualization: J.L. and Y.W.; performing experiments and analyzing data: Y.W.; writing—original draft preparation: Y.W.; writing—review and editing: Y.W., X.W., Z.D., Q.Q., F.X., Z.Q.; funding acquisition: W.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Joint Funds of the National Natural Science Foundation of China under Grant No. U2003208.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CNN | Convolutional Neural Network |
| LSTM | Long Short-Term Memory |
| Bi-LSTM | Bi-directional Long Short-Term Memory |
| GCN | Graph Convolutional Network |

## References

1. Fader, A.; Soderland, S.; Etzioni, O. Identifying relations for open information extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Scotland, UK, 27–31 July 2011; pp. 1535–1545.
2. Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; Etzioni, O. Open information extraction from the web. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007.
3. Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; Zhao, J. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 221–231.
4. Sorokin, D.; Gurevych, I. Context-aware representations for knowledge base relation extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1784–1789.
5. Fan, R.S.J.J.Y.; Chua, T.H.C.T.S.; Kan, M.Y. Using syntactic and semantic relation analysis in question answering. In Proceedings of the 14th Text REtrieval Conference (TREC), Gaithersburg, MD, USA, 15–18 November 2005.
6. Yih, W.T.; He, X.; Meek, C. Semantic parsing for single-relation question answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 23–25 June 2014; pp. 643–648.
7. Socher, R.; Huval, B.; Manning, C.D.; Ng, A.Y. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 1201–1211.
8. Hashimoto, K.; Miwa, M.; Tsuruoka, Y.; Chikayama, T. Simple customization of recursive neural networks for semantic relation classification. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1372–1376.
9. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
10. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 31 May–5 June 2015; pp. 39–48.
11. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 207–212.

12. Xu, K.; Feng, Y.; Huang, S.; Zhao, D. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv* **2015**, arXiv:1506.07650.
13. Kambhatla, N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, Barcelona, Spain, 21–26 July 2004; p. 22.
14. Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; Jin, Z. 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, Lisbon, Portugal, 17–21 September 2015; pp. 1785–1794.
15. Cai, R.; Zhang, X.; Wang, H. Bidirectional recurrent convolutional neural network for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 756–765.
16. Liu, Y.; Wei, F.; Li, S.; Ji, H.; Zhou, M.; Wang, H. A dependency-based neural network for relation classification. *arXiv* **2015**, arXiv:1507.04646.
17. Miwa, M.; Bansal, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv* **2016**, arXiv:1601.00770.
18. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
19. Chen, Z.M.; Wei, X.S.; Wang, P.; Guo, Y. Multi-label image recognition with graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5177–5186.
20. Li, L.; Gan, Z.; Cheng, Y.; Liu, J. Relation-aware graph attention network for visual question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 10313–10322.
21. Babič, M.; Mihelič, J.; Calì, M. Complex network characterization using graph theory and fractal geometry: The case study of lung cancer DNA sequences. *Appl. Sci.* **2020**, *10*, 3037. [CrossRef]
22. Li, G.; Muller, M.; Thabet, A.; Ghanem, B. Deepgcns: Can gcns go as deep as cnns? In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9267–9276.
23. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, Hong Kong, 4–9 December 2017; pp. 5998–6008.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
26. Kim, M.; Park, S.; Lee, W. A Robust Energy Saving Data Dissemination Protocol for IoT-WSNs. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 5744–5764.
27. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
28. Guo, Z.; Zhang, Y.; Teng, Z.; Lu, W. Densely connected graph convolutional networks for graph-to-sequence learning. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 297–312. [CrossRef]
29. Zhang, Y.; Qi, P.; Manning, C.D. Graph convolution over pruned dependency trees improves relation extraction. *arXiv* **2018**, arXiv:1809.10185.
30. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
31. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware attention and supervised data improve slot filling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 35–45.
32. Santoro, A.; Raposo, D.; Barrett, D.G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; Lillicrap, T. A simple neural network module for relational reasoning. *arXiv* **2017**, arXiv:1706.01427.
33. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075.
34. Zhang, Y.; Guo, Z.; Lu, W. Attention guided graph convolutional networks for relation extraction. *arXiv* **2019**, arXiv:1906.07510.