ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200360

A Weighted GCN with Logical Adjacency Matrix for Relation Extraction

Li Zhou[1] and **Tingyu Wang**[1] and **Hong Qu**[2] and **Li Huang** and **Yuguo Liu** [3]

Abstract. Graph convolutional network (GCN), with its capability to update the current node features according to the features of its first-order adjacent nodes and edges, has achieved impressive performance in dependency capturing. But some important nodes from which we should figure out the dependencies are not first-order reachable, which calls for multi-layer GCNs for indirect relevance capturing. In this paper, we propose a novel weighted graph convolutional network by constructing a logical adjacency matrix which effectively solves the feature fusion of multi-hop relation without additional layers and parameters for relation extraction task. And we apply an Entity-Attention mechanism to enrich the entity pairs with more focused semantic information. Experimental results on TACRED and SemEval 2010 task 8 show that our model can take better advantage of the structural information in the dependency tree and produce better results than previous models.

1 INTRODUCTION

Relation extraction aims to capture semantic relations between marked entity pairs in unstructured sentences, which plays a significant role in natural language processing downstream tasks, such as question answering [27], relation inference [31], biomedical knowledge discovery [19], etc. Extracted relation usually occurs between two or more entities of a certain type (e.g. Person, Organisation, Location) and falls into a number of semantic categories (e.g. married to, employed by, lives in). A good relation extraction model facilitates an in-depth semantic understanding of the text content.

Most existing relation extraction models are based on deep learning such as RNN, CNN and their improved models. A relation extraction model takes the text sequence as input, obtains the word-level representation or sentence-level semantic representation through a specifically designed feature extractor, and finally acquires the relation between entities through a classifier. When extracting the relation between entities, predicates are usually of great significance, which means that long distance between entity and predicate is very likely to cause key information loss. To handle this problem, dependency trees [9] were proposed to capture long-distance semantic dependencies and simplify complex sentences for core content extraction. The root of the dependency tree is mostly the predicate of the sentence, and the rest of the main words are centered around the predicate, shown as Figure 1.

For better capturing of the most relevant information, early models apply neural networks to the shortest dependency paths between entities. SDP-LSTM [25] applies LSTM to the sequence of words in the shortest path, DepNN [13] applies RNN to extract subtree features and CNN to extract shortest path features. And Miwa et al. [16] reduced the dependency tree to subtrees under the lowest common ancestor (LCA) between entities. However, these models, running directly on a dependency tree and having difficulty in parallelization, are computationally inefficient because it's often not easy to align trees for efficient batch training.

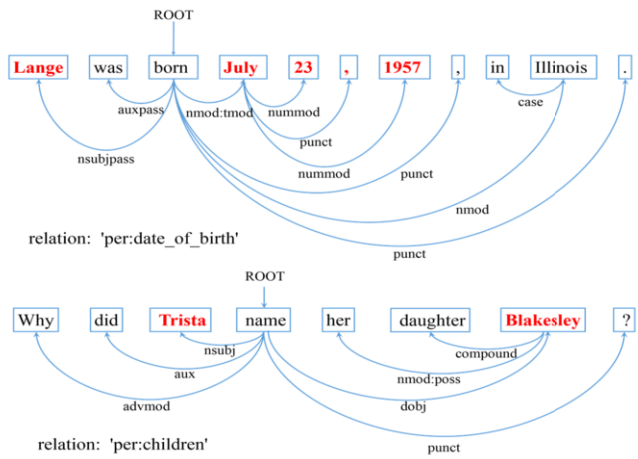

Figure 1: Dependency trees of two samples in TACRED. The curve denotes dependency relation, and the type of the dependency relation is marked on the curve, the predicate is the root of the dependency tree under normal conditions, and the bold red words are entities from which to extract a certain relation in a given sentence.

There are many non-Euclidean data structures like dependency tree on which the performance of CNN and LSTM is very limited because they are often used to process Euclidean data. Kipf and Welling [10] proposed a graph convolutional network (GCN) which makes non-Euclidean data processing possible and has very broad prospects in applications that depend on dependency information modality.

As for relation extraction task, Zhang et al. [29] proposed an extension of GCN, which can be effectively paralleled on any dependency tree structures. They also proposed a pruning strategy that preserves some important words (e.g. "not") that are not on the shortest path between two entities. Besides, GCN-based models have also achieved breakthroughs in other NLP tasks, such as Semantic Role Labeling [15], Neural Machine Translation [1], Multi-Document Summarization [26].

However, the efficiency and effectiveness of existing GCN-based

[1] Li Zhou and Tingyu Wang made the equal contribution.
[2] Corresponding author: Hong Qu (e-mail: hongqu@uestc.edu.cn).
[3] Li Zhou, Tingyu Wang, Hong Qu, Li Huang and Yuguo Liu are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

models are limited in only establishing the first-order[4] model dependence between words. Some important words (e.g. "23" in Figure 1) are linked with their predicate indirectly. So it is necessary to stack multi-layer GCN if k-order neighborhood dependence is needed. Empirically, neural networks with deep structures and more parameters can produce better experimental results. Although GCN can have significant advantages over other methods, it has some fundamental drawbacks. Li et al. [12] showed that GCN brings potential concerns of over-smoothing with many convolutional layers.

To address these problems, we propose a novel weighted graph convolutional network model (WGCN) for relation extraction. In the proposed model, we add virtual edges to the dependency tree to construct a logical adjacency matrix (LAM), which can directly figure out k-order neighborhood dependence with only 1-layer WGCN. We utilize residual blocks [7] between layers of WGCN to alleviate the vanishing gradient. We also apply an Entity-Attention(EA) mechanism to enrich entity representation with more focused between-words semantic information, which facilitates relation extraction of entity pairs.

We evaluate the performance of model on two datasets: the popular SemEval 2010 Task 8 dataset [8] and the more recent, larger TACRED dataset [30]. Our model achieves a delightful performance on both datasets without loss of computational efficiency. Our code is available at `https://github.com/LILI-ZHOU/EA-WGCN`.

Our main contributions are summarized as follows:

- We propose a novel Weighted Graph Convolutional Network (WGCN) model that can obtain k-order neighborhood information on only 1-layer network without additional network parameters, and we alleviate the vanishing gradient problem in graph network by introducing residual blocks.
- We propose Entity-Attention mechanism (EA) to enrich entity representation with more relevant information.
- Finally, we analyze the highlights and complementary effects of LSTM, attention mechanism and GCN in natural language processing.

2 RELATED WORK

The methods of relation extraction can be divided into four categories: supervised, semi-supervised, weakly supervised and unsupervised. Methods for Supervised relation extraction are mainly feature-based and kernel-based. Zhou et al. [4] used SVM as a classifier to study the influence of lexical, syntactic and semantic characteristics on relation extraction task. The supervised method requires manual annotation for a large amount of training data, which is time-consuming and effort-wasting. Hence, the relation extraction methods based on semi-supervision, weak supervision and unsupervision were proposed to solve the problem of the arduous manual annotation works. Brin S [2] presented a technique to grow the target relation from a small sample by taking advantages from the duality between sets of patterns and relations. Craven et al. [3] first proposed a weakly supervised method to extract structured data from texts and build a biological knowledge base. Hasegawa et al. [5] started the pilot work with an unsupervised method for extracting relation between entities. These classical methods have the problem of error propagation in feature extraction, which greatly undermines the performance of relation extraction.

[4] "first-order" means the neighbor nodes to which the target node only need 1 step, and "k-order" requires steps within distance k.

With the popularity of deep learning, scholars gradually apply deep neural networks to relation extraction tasks [11]. Compared with the classical relation extraction methods, the main advantage of the deep-learning-based relation extraction method is that the neural network model can automatically learn sentence features without complex feature engineering. Originally, the relation extraction methods based on deep learning tend to choose structures such as RNN, CNN and their improved models. With the appearance of graph convolutional network (GCN) [10], some GCN-based models have come into being.

Relation extraction model based on RNN. The method of relation extraction based on RNN model was first proposed by Socher et al. [22] in 2012. This method assigns a vector and a matrix to every node in a parse dependency tree: the vector captures the inherent meaning of the constituent, while the matrix captures how it changes the meaning of neighborhood words or phrases. Hashimoto et al. [6] proposed a recursive neural network (RNN) model based on syntactic tree in which POS tags, phrase categories and syntactic head are also adopted. Traditional RNN has difficulty in dealing with long-term dependence, while LSTM (an advanced RNN structure with long short term memory) solves these problems by adding a cell state and three gated operations. Xu et al [25] leverages the shortest dependency path (SDP) between two entities and applies multi-channel LSTM unites to pick up heterogeneous information along the SDP.

Relation extraction model based on CNN. Zeng et al [28] exploit a convolutional deep neural network to extract lexical and sentence-level features. Their method takes all of the word tokens as input without complicated pre-processing. Xu et al [24] proposed a relation extraction model of convolutional neural network based on dependency tree. The difference between this model and the CNN model of Zeng et al [28] is that the input text pass the dependency tree in the former model.

Relation extraction model based on GCN. Kipf and Welling [10] presented a GCN model for semi-supervised learning on graph-structured data that is based on an efficient variant of convolutional neural networks which operate directly on graphs. Zhang et al. [29] applied GCN to the dependency tree for relation extraction, which pools information over arbitrary dependency structures efficiently in parallel. They also created a pruning strategy to the input trees by keeping words immediately around the shortest path between the two entities where may lies a relation.

Our model is also based on GCN, but existing GCN, only able to obtain the first-order neighborhood information directly, needs multi-layer structure to figure out k-order neighbor information indirectly. Considering this limitation, we build a novel Weighted Graph Convolutional Network (WGCN) by adding virtual edges on the graph structure, which can directly obtain the k-order neighborhood information. In this way, while retaining the inherent advantages of GCN, the accuracy of the model is improved without losing simplicity.

3 MODEL

We first define the task of relation extraction. χ= [x_1,$\mathrm{x}_2 \ldots \mathrm{x}_N$] represents a sentence, where x_i is the i^{th} token, N is the length of χ. And $\chi_s = [\mathrm{x}_{s_1} \ldots, \mathrm{x}_{|\chi_s|}]$, $\chi_o = [\mathrm{x}_{o_1} \ldots, \mathrm{x}_{|\chi_o|}]$ denote the subject entity span and the object entity span respectively in the sentence. Given χ, χ_s and χ_o, relation extraction is to make a prediction of the relation $r \in R$ between the two entities. R represents a predefined relation set.

In this section, we will introduce our novel model (EA-WGCN)

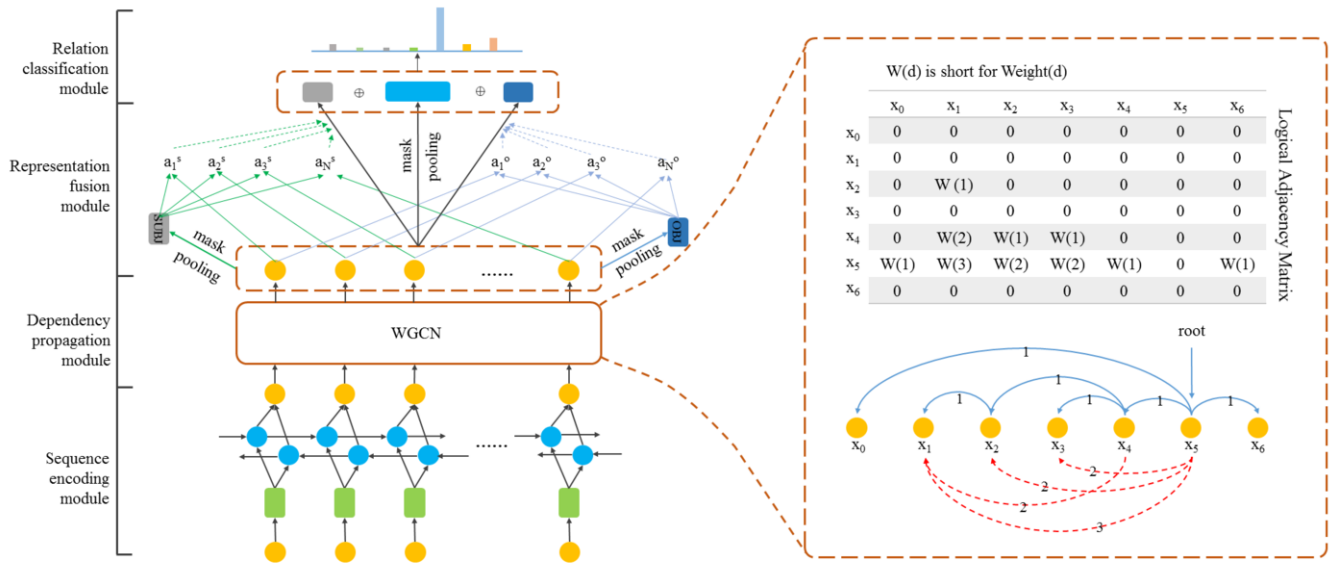

Figure 2: Relation extraction with EA-WGCN. The left part shows the overall architecture of EA-WGCN, and the right part shows the logical adjacency matrix for the WGCN. The model is composed of four modules. The sentences are firstly fed into the Sequence encoding module to get contextual and sequential representation as input for WGCN. The Sequence encoding module contains an embedding layer and a Bi-LSTM layer. In the WGCN layer, an advanced logical adjacency matrix was constructed by adding virtual edges on the sentence dependency tree. The final sentence representation and entity representation are obtained through Entity-Attention mechanism and max-pooling in the Representation fusion module, which are concatenated as the input of the final relation classifier. The details of Logical Adjacency Matrix is demonstrated in the right part, the following is a sentence dependency tree. The blue solid edge represents the direct dependency of adjacent nodes in the sentence. When two nodes are reachable throw an accessible path but not directly connected by a solid edge, we add a red dotted edge between them to represent an indirect dependency. The number on all edges represents the shortest path length between nodes, which is reflected in the elements of the logical adjacency matrix.

with a weighted graph convolutional network structure and an Entity-Attention mechanism, which can better capture the structural information in the dependency tree of a sentence and produce a better result for relation extraction task. Figure 2 illustrates the overview of the model. The model mainly consists of four modules including (1) Sequence encoding module (2) Dependency propagation module (3) Representation fusion module (4) Relation classification module. The innovation of our model is mainly reflected in the second and third modules.

3.1 Sequence Encoding Module

This module mainly consists of an embedding layer and a bidirectional LSTM layer (Bi-LSTM). In embedding layer, the word embedding, NER label embedding and POS tag embedding of each token are concatenated as follows:

$$\mathrm{e}_t = [{e_t}^{word} : {e_t}^{ner} : {e_t}^{pos}] \in \mathbb{R}^m \tag{1}$$

$$m = d_{word} + d_{ner} + d_{pos} \tag{2}$$

where d_{word}, d_{ner}, d_{pos} denote the dimension of word, NER, POS embedding, e_t is the concatenated representation vector of a token at time step t. The vectors of all time steps are serialized into a 2-D matrix $E = [e_1, e_2, \ldots, e_N] \in \mathbb{R}^{N \times m}$. The vectors in E are merely independently juxtaposed word-level representation with little sentence-level information. To obtain contextual semantic representations, we concatenate both the forward LSTM state and the backward LSTM state in a Bi-LSTM layer, shown as follows:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(x_t, h_{t-1}) \in \mathbb{R}^{d_l} \tag{3}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(x_t, h_{t+1}) \in \mathbb{R}^{d_l} \tag{4}$$

$$\overrightleftarrows{h_t} = [\overrightarrow{h_t} : \overleftarrow{h_t}] \in \mathbb{R}^{2d_l} \tag{5}$$

where d_l denotes the LSTM hidden dimension, and $\overrightleftarrows{h_1}, \overrightleftarrows{h_t}, \ldots, \overrightleftarrows{h_N}$ as the output of sequence encoding module has already contained bidirectional semantic feature.

3.2 Dependency Propagation Module

Graph convolutional network. Before introducing this module, we review Graph Convolutional Network (GCN) [10], which provides a new method for processing graph-structured data. Given $G = (V, E)$,the input of GCN is:

- A feature matrix X ,whose shape is $N \times F^0$,where N represents the number of nodes in the graph, F^0 is the input feature dimension of each node.
- A $N \times N$ adjacency matrix A of this graph, where A_{ij}=1 if there is an edge going from node i to node j.

Hence, the output of l-layer GCN is written as:

$$H^{(l)} = \sigma(AH^{(l-1)}W^{(l)} + b^{(l)}) \tag{6}$$

where $H^0 = X$, W^l is a linear transformation, b^l is a bias term, and σ is a nonlinear function (e.g.,RELU).

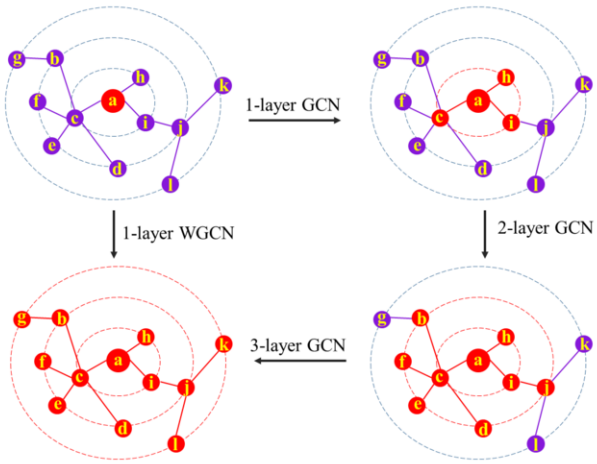

Figure 3: Comparison of WGCN with vanilla GCN. Take the node a of the up-left part as an example. At the beginning, node a only contains its own feature. After 1-layer GCN, as shown in the up-right part, node a acquires the features of its neighborhood nodes c, h and i. At the same time, node c is also updated by the features of its neighbors, so do node h and i. And after 2-layer GCN, as depicted in the down-right part, node a gets the updated features of its neighborhood nodes c, h and i again. Since the features of nodes c, h and i already contain the features of their neighbors after the previous GCN, node a indirectly obtains the features of the neighborhood nodes of node c, h and i. Thus, after 3-layer GCN, node a is updated with information from all nodes directly and indirectly connected to it, which is vulnerable to over-smoothing. For WGCN, we add virtual edges between node a and nodes indirectly connected to it. Hence, after 1-layer WGCN, node a can obtain the features of all nodes with paths to it, shown as the down-left part.

On the relation extraction task, we parse a dependency tree as a graph structure on the sentence in which each token represents a node. And if there is a dependency between words, there is an edge between corresponding nodes. After each graph convolutional operation, information of each node can be updated by fusing the feature of its neighborhood nodes.

Weighted graph convolutional network. However, feature fusion on 1-layer GCN only represents first-order neighborhood dependency. When k-order neighborhood feature is required for further relation extraction work, it can only be indirectly acquired through multi-layer GCN structure, which is time-consuming and has high tendency of over-smoothing, shown as Figure 3.

To avoid this limitation and realize the multi-hop feature fusion in a single layer graph network, we propose a Weighted Graph Convolutional Network(WGCN). In the proposed model, we add virtual edges to the dependency tree to construct a logical adjacency matrix (LAM), which can directly figure out k-order neighborhood dependence with only 1-layer WGCN. The algorithm of constructing LAM is shown as Algorithm 1.

Algorithm 1 Obtain Logical Adjacency Matrix (LAM)

Input: T: dependency tree of sentence; N: the sequence length;
Output: LAM

1: **Initial** $LAM \in \mathbb{R}^{N \times N}$, all elements in LAM are zero;
2: **Traverse** each node i from the root of T:
3: Traverse all nodes j in the subtree whose root is node i;
4: Compute the distance d between node i and node j;
5: Set $LAM_{ij} = Weight(d)$

The $Weight$ function in the Algorithm 1 is used to calculate the weight coefficient of feature fusion between nodes. The shorter the distance between nodes, the greater the weight, and vice versa. The fusion weight coefficient between adjacent nodes is 1, meaning the maximum information fusion weight. In our model, we choose the $Weight$ function defined as:

$$Weight(d) = \frac{1}{e^{d-1}} \tag{7}$$

Where e is the Euler's number. Then we can obtain a new propagation formula for the fusion of dependent information shown as follows:

$$h_i{}^{(l)} = \sigma(\sum_{j=1}^{N} \widetilde{LAM}_{ij} h_j{}^{(l-1)} W^{(l)} / d_i + b^{(l)}) \in \mathbb{R}^{d_w} \tag{8}$$

where $h_i{}^0 = \overrightleftharpoons{h_i}$ in Equation 5, $\widetilde{LAM} = LAM + I$,which means all nodes in dependency tree added self-loop connection, and $d_i = \sum_{j=1}^{N} \widetilde{LAM}_{ij}$ is the degree of node i, d_w denotes the WGCN hidden size, and $W^{(l)} \in \mathbb{R}^{2d_l \times d_w}$.

In this way, 1-layer WGCN can integrate the k-order neighborhood information directly without extra parameters introduced. The gradient of the graph network gradually disappears as the depth increases, which makes the receptive field of WGCN very likely to produce a lot of noise in the process of information transmission. Hence, the residual blocks [5] are built to alleviate this problem in WGCN. Through this module,we can obtain the dependency representation of sentence.

3.3 Representation Fusion Module

To represent the subject entity and the object entity in the sentence with more focused semantic information, we propose to design an Entity-Attention mechanism(EA) under which an entity can capture its correlated parts of sentence. Firstly, the entity representation without attention is defined as:

$$h'_{entity} = maxpool[H^{(L)}_{es:ee}] \tag{9}$$

where $H^{(L)} = [h_1{}^{(L)}, h_2{}^{(L)}, \ldots, h_N{}^{(L)}]$ is matrix representation of sentence after L-layer WGCN, es indicates the start subscript of the entity and ee indicates the end subscript. The $maxpool$ function reduces the representation from 2-dimension to 1-dimension as d_w. Then the final entity representation with Entity-Attention is given by:

$$a = softmax(H^{(L)} h'_{entity}) \tag{10}$$

$$h_{entity} = maxpool[(aH^{(L)})_{es:ee}] \tag{11}$$

where a is a vector of entity-to-sentence attention weights. Then the final representation h_s, h_o of χ_s,χ_o can be obtained from Equation 9,10,11, which are already enriched with focused information. And we also obtain the sentence representation vector directly by:

$$h_{sent} = maxpool[H^{(L)}] \tag{12}$$

Finally, we integrate all the features by concatenating the final representations of sentence and entities as follows[21]:

$$h_{out} = [h_{sent}; h_s; h_o] \tag{13}$$

[5] We use the output of the last-layer WGCN directly as the output of the dependency propagation module.

3.4 Relation Classification Module

In this module, the final fusion representation containing abundant sequential and dependency information of the original text is fed into a feed-forward neural network with a softmax operation in which we can get a probability distribution over relations.

This model can be trained by backpropagation and the cross entropy function is used as the loss function of the model during training. Our competitive advantage lies in that we have achieved better performance without extra parameters and complex structure.

4 EXPERIMENT

4.1 Datasets

We evaluate the performance of our model on two relation extraction datasets: TACRED and SemEval 2010 Task 8.

- **TACRED.** The TACRED dataset [6] is a large-scale relation extraction dataset consists of 106,264 samples and 42 relation types(including 41 defined types and a special relation label 'no relation' if no defined relation is held) [30]. The content is mainly the text corpus of newswire and TAC Knowledge Base Population (TAC KBP) challenges. In each TACRED example, the following annotations are provided: the spans of the subject and object mentions; the types of the mentions (among 23 fine-grained types used in the Stanford NER system); the 42 types of relation held between the entities.
- **SemEval 2010 Task 8.** The SemEval 2010 Task 8 is a public dataset which contains 10,717 instances with 9 relations and a special 'Other' class which means that the relation does not belong to any of the nine relation types. To parse this original data, we use Stanford CoreNLP [14] to generate dependency trees, POS, and NER sequences.

For both datasets, we use pre-trained 300-dimensional GloVe [18] to initialize word embedding, and randomly initialized POS embedding and NER embedding with 30 dimensions.The Bi-LSTM hidden size is set to 100, and WGCN hidden size is set to 200, which effectively enables residual computation. And we set the dropout rate 0.5, prune $k = 1$ [29]. For TACRED, We choose 2 layer WGCNs, initializing learning rate 1.0 with a decay rate 0.9. For SemEval, we set 3 layer WGCNs, 0.5 learning rate with a decay rate of 0.95. For both datasets, we trained our model for 150 epochs.

4.2 Evaluation

We use precision(P), recall(R) and F1 score(F1) to evaluate our models.

We follow the convention and report the official micro-averaged F1 scores on TACRED dataset. The official evaluation metric uses micro-averaged F1 over instances with proper relationships (excluding the "no-relation" type).

On SemEval, we test the model performance using the official scorer in terms of the macro-F1 score over the nine relation pairs. However, the "other" class is not taken into consideration when we compute the official measures.

By the way, 'micro' means to calculate metrics globally by counting the total true positives, false negatives and false positives. The corresponding elements (TP, FN, FP, TN) in each confusion matrix were averaged respectively, and then the micro-precision and micro-recall were obtained to calculate micro-F1.While "macro" means to calculate metrics for each label, and find their unweighted mean. Precision and recall are calculated for each confusion matrix respectively, so as to obtain macro-precision and macro-recall, and then macro-F1 is calculated.

[6] https://nlp.stanford.edu/projects/tacred/

4.3 Results

TACRED. The experimental results of TACRED in Table 1 show that our model EA-WGCN outperforms all compared models. We mainly compare our model with the following four types of models: 1) Traditional relation extraction model: a logistic regression model (LR), which combines dependency tree information with other lexical information. 2) CNN-based relation extraction model: Nguyen et al. [17] depart from these traditional approaches with complicated feature engineering, and apply a Convolutional Neural Network model (CNN), that automatically learns features from sentences through multiple window sizes for filters. 3) LSTM-based relation extraction model: Position-aware LSTM (PA-LSTM) [30] which combines a LSTM Sequence model with a form of entity position-aware attention; Shortest Dependency Path LSTM (SDP-LSTM) [25] which applies LSTM to the shortest dependent path between entities; tree-structured LSTM (tree-LSTM) [23], a generalization of LSTMs to tree-structured network topologies. 4) GCN-based relation extraction model: contextualized graph convolutional network (C-GCN) presented by Zhang et al. [29], which applies graph convolutional network in the pruned dependency tree.

By observing the experimental results, we find that our model improved at least 1.2 F1 compared with other models. CNN achieves the highest precision score 75.6 and a lowest recall score 47.5, which leads to a lowest F1 score. We hypothesize that CNN may tend to precisely classifying the defined relations while make misclassification between defined and undefined types. And our EA-WGCN model obtains the highest recall score 64.8 and the highest F1 67.6. In particular, compared to the C-GCN, our model has a certain improvement in precision score, recall score and $F1$ score without extra layers and parameters. We also run an ensemble of our EA-WGCN model by averaging the softmax results of 5 randomly initializations, which improves the $F1$ score by 1%.

System	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
Our Model(EA-WGCN)	70.8	**64.8**	**67.6**
Our Model(ensemble)	71.3	**66.1**	**68.6**

Table 1: Results on TACRED. Comparative experimental results are mainly reported from Zhang et al. $+$ remarks result quoted from [29],and $++$ remarks result quoted from [30].

SemEval 2010 Task 8. To demonstrate the versatility of our proposed model, we have also conducted experiments on another

dataset——SemEval. We experimented mainly with some dependency models, shown as Table 2. SemEval dataset is much smaller than TACRED dataset, but our model still obtained F1 85.1 and outperformed any other dependency models. In the same ensemble approach, we elevated the F1 score of our single EA-WGCN model to 85.4.

System	F_1
SVM$^+$ [20]	82.2
DepNN$^+$ [13]	83.6
SDP-LSTM$^+$ [25]	83.7
SPTree$^+$ [16]	84.4
C-GCN^{++}	<u>84.4</u>
C-GCN$^+$ [29]	84.8
Our Model(EA-WGCN)	<u>85.1</u>
	85.4*

Table 2: Results on SemEval.+ indicates results are reported in the original papers in which the methods are proposed, ++ indicates results are generated with our implementation. The underline indicates that results produced from single models, and $*$ represents results of single models ensemble.

5 ANALYZE & DISCUSSION

5.1 Ablation study

To prove the contribution of each component of our model, we ran ablation studies on them. On TACRED dataset, we used an experiment in which EA-WGCN model scored F1 67.9 on the validation set as the standard. During the ablation experiment, the parameters of each experiment were initialized from the same setting of random seeds, so as to ensure the fairness of the experiment. This result is shown in Table 3. We find that 1) When LAM is replaced by the ordinary adjacency matrix, F1 score drops by 0.5, 2) When we remove Entity-Attention mechanism, the score of F1 drops by 1.0, 3) When we remove the residual block, F1 score drops by 1.1, which proves that the residual block can effectively alleviate the vanishing gradient problem in deep graph network, 4) When we remove the Bi-LSTM layer, F1 score drops by 6.0. 5) When we change the LAM to unit matrix I[7], F1 score drops by 2.4.

Model	Dev F1
EA-WGCN	**67.9**
-Logical adjacency matrix (LAM)	67.4
-Entity attention (EA)	66.9
-Residual block	66.8
-Bi-LSTM layer	61.9
-WGCN layer	65.5

Table 3: Aablation study on TACRED

[7] At this point, the WGCN layer down grades into a full connection layer.

5.2 Effect of Logical Adjacency Matrix

In our study, we insisted that the Logical Adjacency Matrix (LAM) can capture k-order neighborhood information with only 1-layer GCN, without adding additional layers and parameters. In order to prove the effectiveness of LAM in relation extraction model, we conducted a comparative experiment between EA-WGCN (our model) and EA-GCN in which the LAM in our model is replaced by an ordinary adjacency matrix. Convergence results under different Adjacency Matrix strategies are shown in Figure 4. Our model quickly converged to a virtually better solution. EA-GCN also performed quite well, albeit with a slower converge rate than our model. In this case, we also compare the best dev F1 scores between EA-GCN and our model, which is shown in figure 5. In terms of final best F1 score, our model outperformed the EA-GCN by at least 0.5 F1 score and reached a peak around the time of the 40th epoch. The above has proved that LAM can effectively capture k-order neighborhood features and obtain better prediction results in relation extraction task.

5.3 Effect of Entity-Attention

It is enlightening to analyze the significance of each sentence word in determining entity representation for relation extraction between entities.

Figure 5 manifests the influential extent of each word in the sentence on a given entity. The color depth indicates the importance degree of the weight in attention vector a in Equation 10, the darker the more important. Take the first sentence for example, the relation between the subject ("The Federation") and the object ("1994") is "org:founded". We observed that the Entity-Attention mechanism leads the two entities to pay more attention to the phrase "founded in 1994". Similarly, the entities of the second sentence are more concerned with the phrase "12500 employees in", which leads our model to extract the relation ' org:number_of_employees/members '. Obviously Entity-Attention can help to get importance-guided entity representation from the whole sentence dynamically.

5.4 Analyze of LSTM & Attention & GCN

In natural language processing models based on deep learning, LSTM and attention mechanism are widely used. By concatenating forward and backward LSTM state, each word in the text can obtain a representation with contextual semantics. Attention mechanism can help to focus more on important parts and less on other unimportant factors. In our model, entities are used as query vector to assign attention weight to each word in a sentence through Entity-Attention mechanism, which plays a role of global observation. The latest GCN model based on sentence dependency tree allows each word to directly capture the information of its dependent words even far away from it in the original text. Therefore, LSTM, attention mechanism and GCN have different emphasis on feature extraction.Shown as Table 3, all three feature extractors contribute F1 score to our model , which illustrates the complementary effects of LSTM in sequential information capturing, attention mechanism in global relevance obtaining, and GCN in dependency acquiring. Combining LSTM, attention mechanism and GCN enriches word-level and sentence-level representation with more abundant information to capture as much semantic information as possible, which can achieve more accurate relation extraction.

Besides, LAM in our WGCN is an N*N matrix. And for each word, it calculates a fusion weight coefficient for other words, and

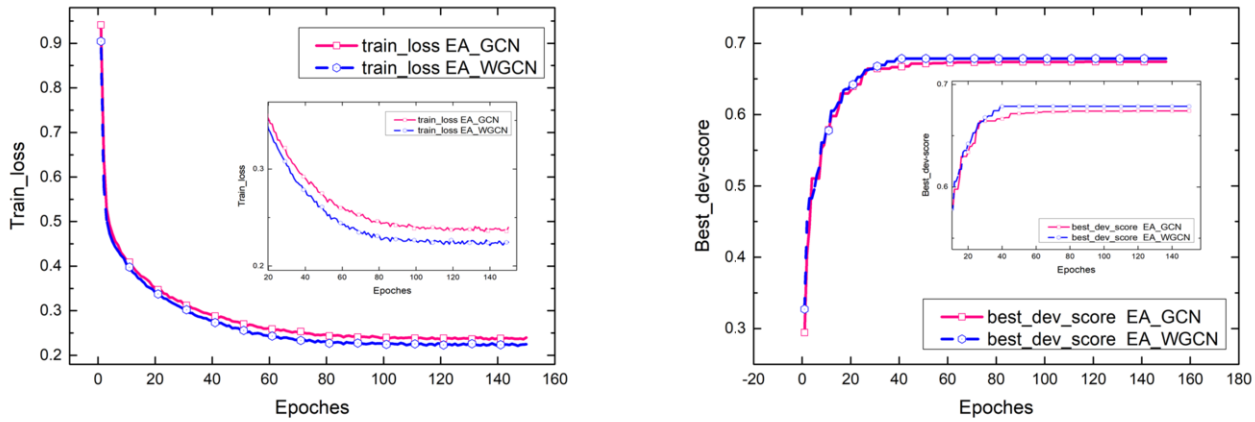

Figure 4: Performance of GCN-based models under different Adjacency Matrix strategies. For each model we show the training loss and best dev F1 score on the TACRED train and dev set. Our model outperforms EA-GCN without a logical adjacency matrix.

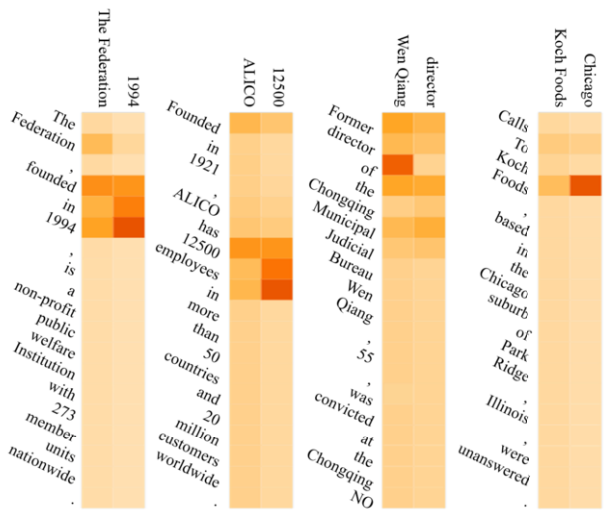

Figure 5: Entity-Attention Visualization. The color depth expresses the importance degree of the weight in attention vector a in Equation 7. From left to right, the relations of the four examples in TACRED are: (1) org:founded; (2) org:number_of_employees/members; (3) per:title; (4) org:city_of_headquarters. On the left of each strip is a complete sentence, with the subject entity and object entity on.

the sum of the weights are normalized to 1, which looks very similar to the attention mechanism. But in fact, LAM only contains relevant information of words which are directly or indirectly reachable in the sentence dependency tree, which is very helpful for relation extract task, while attention mechanism focuses on global relationships among all words in the sentence.

6 CONCLUSION

In this paper, we introduce the novel Weighted Graph Convolutional Network with Entity-Attention mechanism (EA-WGCN) for relation extraction. In WGCN, we construct a logical adjacency matrix by adding virtual edges between nodes that have paths but are not directly adjacent to each other in the dependency tree. Such operations can directly obtain k-order neighborhood information through only 1-layer WGCN, which enables multi-hop relation with a simple structure. By introducing residual blocks between WGCN layers, the vanishing gradient problem is effectively alleviated. And Entity-Attention mechanism enables entity representation to obtain importance-guided semantic information from sentences. Experimental results on both TACRED dataset and SemEval 2010 task 8 dataset show that EA-WGCN can make a more comprehensive use of the structural information in the dependency tree and produce better results than previous models. We also find the complementary effects of LSTM in sequential information capturing, attention mechanism in global relevance obtaining, and GCN in dependency acquiring.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation of China under Grant 61573081 and Grant 6180604.

REFERENCES

[1] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an, 'Graph convolutional encoders for syntax-aware neural machine translation', *arXiv preprint arXiv:1704.04675*, (2017).

[2] Sergey Brin, 'Extracting patterns and relations from the world wide web', in *International Workshop on The World Wide Web and Databases*, pp. 172–183. Springer, (1998).

[3] Mark Craven, Johan Kumlien, et al., 'Constructing biological knowledge bases by extracting information from text sources.', in *ISMB*, volume 1999, pp. 77–86, (1999).

[4] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min, 'Exploring various knowledge in relation extraction', in *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 427–434. Association for Computational Linguistics, (2005).

[5] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman, 'Discovering relations among named entities from large corpora', in *Proceedings of the 42nd annual meeting on association for computational linguistics*, p. 415. Association for Computational Linguistics, (2004).

[6] Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama, 'Simple customization of recursive neural networks for semantic relation classification', in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1372–1376, (2013).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, (2016).

[8] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz, 'Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals', in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pp. 94–99. Association for Computational Linguistics, (2009).

[9] Nanda Kambhatla, '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*, p. 22. Association for Computational Linguistics, (2004).

[10] Thomas N Kipf and Max Welling, 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907*, (2016).

[11] Shantanu Kumar, 'A survey of deep learning methods for relation extraction', *arXiv preprint arXiv:1705.03645*, (2017).

[12] Qimai Li, Zhichao Han, and Xiao-Ming Wu, 'Deeper insights into graph convolutional networks for semi-supervised learning', in *Thirty-Second AAAI Conference on Artificial Intelligence*, (2018).

[13] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang, 'A dependency-based neural network for relation classification', *arXiv preprint arXiv:1507.04646*, (2015).

[14] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky, 'The stanford corenlp natural language processing toolkit', in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, (2014).

[15] Diego Marcheggiani and Ivan Titov, 'Encoding sentences with graph convolutional networks for semantic role labeling', *arXiv preprint arXiv:1703.04826*, (2017).

[16] Makoto Miwa and Mohit Bansal, 'End-to-end relation extraction using lstms on sequences and tree structures', *arXiv preprint arXiv:1601.00770*, (2016).

[17] Thien Huu Nguyen and Ralph 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, (2015).

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning, 'Glove: Global vectors for word representation', in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, (2014).

[19] Chris Quirk and Hoifung Poon, 'Distant supervision for relation extraction beyond the sentence boundary', *arXiv preprint arXiv:1609.04873*, (2016).

[20] Bryan Rink and Sanda Harabagiu, 'Utd: Classifying semantic relations by combining lexical and semantic resources', in *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 256–259. Association for Computational Linguistics, (2010).

[21] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap, 'A simple neural network module for relational reasoning', in *Advances in neural information processing systems*, pp. 4967–4976, (2017).

[22] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng, '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*, pp. 1201–1211. Association for Computational Linguistics, (2012).

[23] Kai Sheng Tai, Richard Socher, and Christopher D Manning, 'Improved semantic representations from tree-structured long short-term memory networks', *arXiv preprint arXiv:1503.00075*, (2015).

[24] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao, 'Semantic relation classification via convolutional neural networks with simple negative sampling', *arXiv preprint arXiv:1506.07650*, (2015).

[25] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi 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, (2015).

[26] Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev, 'Graph-based neural multi-document summarization', *arXiv preprint arXiv:1706.06681*, (2017).

[27] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou, 'Improved neural relation detection for knowledge base question answering', *arXiv preprint arXiv:1704.06194*, (2017).

[28] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al., 'Relation classification via convolutional deep neural network', (2014).

[29] Yuhao Zhang, Peng Qi, and Christopher D Manning, 'Graph convolution over pruned dependency trees improves relation extraction', *arXiv preprint arXiv:1809.10185*, (2018).

[30] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning, 'Position-aware attention and supervised data improve slot filling', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, (2017).

[31] Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-seng Chua, and Maosong Sun, 'Graph neural networks with generated parameters for relation extraction', *arXiv preprint arXiv:1902.00756*, (2019).