# Adaptive Graph Convolutional Networks with Attention Mechanism for Relation Extraction

Zhixin Li
*Guangxi Key Lab of Multi-source* Information Mining and Security
*Guangxi Normal University*
Guilin 541004, China
lizx@gxnu.edu.cn

Yaru Sun
*Guangxi Key Lab of Multi-source* Information Mining and Security
*Guangxi Normal University*
Guilin 541004, China
MiniiEcho@163.com

Suqin Tang
*Guangxi Key Lab of Multi-source* Information Mining and Security
*Guangxi Normal University*
Guilin 541004, China
sqtang@gxnu.edu.cn

Canlong Zhang
*Guangxi Key Lab of Multi-source Information Mining and Security*
*Guangxi Normal University*
Guilin 541004, China
clzhang@gxnu.edu.cn

Huifang Ma
*College of Computer Science and Engineering*
*Northwest Normal University*
Lanzhou 730070, China
mahuifang@yeah.net

***Abstract*—In the relationship extraction task of NLP, how to effective use of the rich structural information on the dependency tree is a challenging research problem. To better learn the dependency relationship between nodes, we address the relationship extraction task by capturing rich contextual dependencies based on the attention mechanism, and using distributional reinforcement learning to generate optimal relation information representation. Unlike using an attention mechanism to effectively make use of relevant information, we propose a Dual Attention Graph Convolutional Network (DAGCN) to adaptively integrate local features with their global dependencies. Specifically, we append two types of attention modules on top of GCN, which model the semantic interdependencies in spatial and relational dimensions respectively. The position attention module selectively aggregates the feature at each position by a weighted sum of the features at all positions of nodes internal features. Similar features would be related to each other regardless of their distances. Meanwhile, the relation attention module selectively emphasizes interdependent node relations by integrating associated features among all nodes. We sum the outputs of the two attention modules and use reinforcement learning to predict the classification of nodes relationship to further improve feature representation which contributes to more precise extraction results. The results on the TACRED and SemEval datasets show that the model can obtain more useful information for relational extraction tasks, and achieve better performances on various evaluation indexes.**

***Keywords*—attention mechanism, graph neural network, relation extraction, reinforcement learning**

## I. Introduction

Relation extraction is used to detect relationships between entities in text and plays an important role in natural language processing. Relation extraction is the basis for answering knowledge queries [1], building knowledge graph [2] and also forms an important supporting technology for information extraction. The traditional models focus primarily on the research into entity recognition and rule-based methods. The existing models adopt an end-to-end approach to perform this task more efficiently. Much scholarly research work has shown that the named entity recognition has reached satisfactory result levels, but the rule-based models on poor generalizability. In recent years, some progress has been made in obtaining the relationships between entities using neural network models. This approach trains a model to learn the structural information of sentences without relying on defined rules.

The existing relation extraction models can be divided into two categories: sequence-based and dependency-based. Sequence-based models [3, 4] operate on word sequences, for example, using cyclic neural networks to encode words to obtain the sentence information. Dependency-based models [5, 6] incorporate the dependency tree of the sentence relationship and effectively use the structural information in the dependency tree to extract features. Compared with the sequence-based models, the dependency-based models can capture implicit nonlocal syntactic relationships. However, the information in the dependency tree is not always beneficial to the entity relationship information. Thus, to further improve the system performance, some pruning strategies have been adopted to extract the dependency information. In the relation extraction task, Xu [7] only considered the shortest dependency paths of the entities in the tree, which greatly reduces the data burden. To retain useful and more accurate dependency information, Miwa and Bansal [8] considered only the subtrees of the lowest common ancestor (LCA) of the entity. However, using such pruning strategies risks eliminating some important information concerning the entire tree. As shown in Fig. 1(a), the dependency tree is pruned using the nearest common ancestor strategy. When the furthest dependency path $K$=1 is considered, the implicit information concerning the entities $Gwathmey$ and $Rosalie$ may be missing. To avoid losing this important information and to make better use of the hidden information in the tree, the model should incorporate the entire tree [9], and use an end-to-end approach to learn the strength of the associations between entities. Therefore, the key to the

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

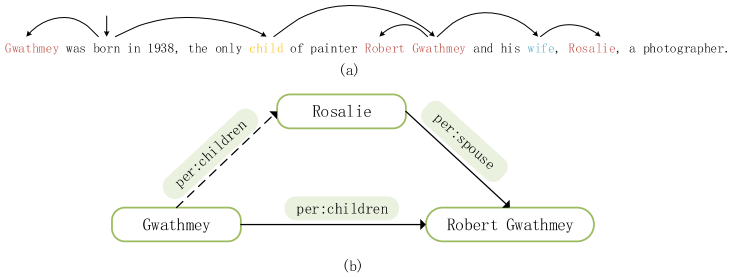


Fig. 1. Relation extraction of a plain text. (a) is the dependency tree of the sentence entity relationship, (b) shows the role of multihop reasoning in text.

task is to make the model learn from the entire tree to maintain a balance between retaining and excluding information.

In addition, the size of the tree may affect the path of the pruning strategy, and multihop reasoning between entities will occur. As shown in Fig. 1(b), an implicit relationship exists between $Gwathmey$ and $Rosalie$ that is based on $Robert\ Gwathmey$. If it that relationship is absent, the implicit relationship will not be established. Multihop relational reasoning is indispensable in multihop relational extraction, thus the key to this task is eliminating the multihop influence on the dependent path.

To solve these two problems, this paper proposes a relationship extraction model that applies a dual attention mechanism with reinforcement learning in a graph convolutional network (DAGCN). A graph neural network (GNN) is an effective approach for solving multihop relational reasoning problems [10]. In this scheme, the words in sentences are represented as nodes in a graph. The node representations depend on their adjacent nodes. The application of a neural network to a graph structure can directly obtain the node dependency information to alleviate the multihop influence on a dependent path. Using a self-attention mechanism [11] not only captures richer semantic information from the text but also enables the model to learn the strengths of the associations between nodes to make better use of the information in the dependency tree. The uncertain information contained in the relationship classification has a great influence on the prediction results. The distributional reinforcement learning is used to consider the uncertain information in the relationship classification, so as to optimize the representation of the relationship information and improve the accuracy of the relationship classification. Therefore, this paper adopts the method of graph neural network with attention mechanism combined reinforcement learning, but different from single attention graph neural network, this method is a double attention graph neural network, which adaptively integrates local features with global dependencies. Specifically, we add two types of attention modules at the last layer of GCN [12]: position attention model and relation attention model. Semantic dependency information is modelled in spatial dimension and relational dimension respectively. we also add a classification reinforcement module at the end of the model to optimize the classification of nodes relationship. For the location attention module, we use the self-attention mechanism to capture the spatial dependency relationship between any two locations of node features. The update of the features at a certain position is obtained by aggregating the features at all positions through a weighted sum operation, where the weight is determined by the similarity of the features corresponding to the two positions. That is any two positions with similar features can promote each other, no matter how far apart in the spatial dimension. For the relation attention module, we use the self-attention mechanism to encode the dependency information between nodes to generate a relation attention matrix and update the features of each node through a weighted sum operation. Finally, we fuse the features of the two attention modules together and by classification reinforcement module generate optimal relationship information representation to further enhance the feature representation. The experimental results show that the proposed model achieves new state-of-the-art performance on the TACRED and SemEval datasets.

Our contributions are summarized as follows.

- We propose a dual attention graph neural network. By capturing a long-range of contextual dependency information and improves the discriminate ability of feature representations in relation extraction.
- We propose a position attention module to learn the spatial correlation of features and also propose a relation attention module to capture the dependency information between nodes. Building wide-range contextual dependent relationships through local features significantly improves the relationship classification results.
- We propose a classification reinforcement module to optimize the representation of relationship features and improve the accuracy of relationship classification. This module use distributional reinforcement learning to consider the uncertain information that affects the classification results into the representation of relational information.
- Our model achieved the new state-of-the-art performance on the TACRED and SemEval datasets.

## II. Dual Attention-guided GCN

In this section, Firstly, we present a general framework of DAGCN and then introduce the two attention modules. Finally, we describe how to aggregate the two attention modules

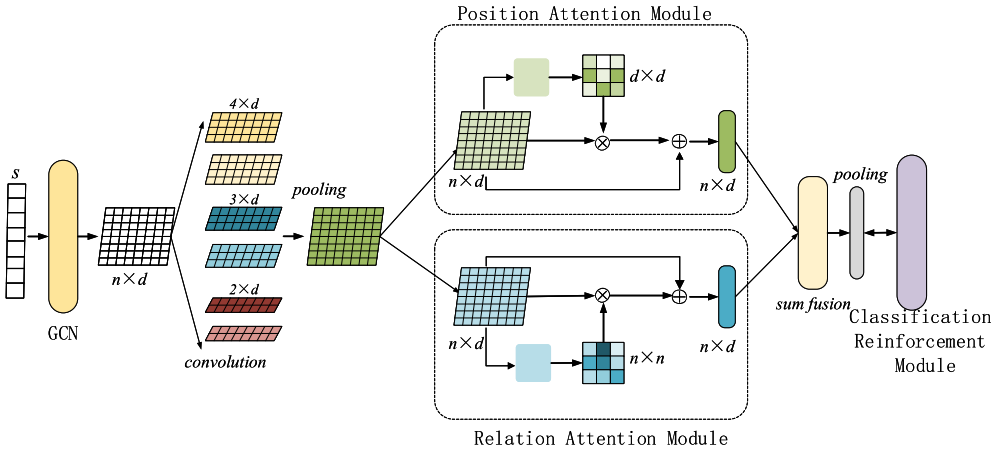


Fig. 2. The proposed DAGCN model.

together and combine the distributional reinforcement learning for further renement.

### A. Overview

Since convolution operations would lead to a local receptive domain, it could have some bad effects for relation extraction task. On the one hand, there may be some differences in the features of nodes with the same label. These differences lead to the inconsistency intra-class and affect the accuracy of recognition. On the other hand, the dependency information between nodes may be lost. To solve these issues, we explore the global contextual information by establishing associations among features with the attention mechanism. This method could adaptively aggregate long-range contextual information, thereby it could improve the representation ability of node feature. Unlike the single attention guidance graph neural network [6] to obtain contextual information, this method not only pays attention to the dependency information between nodes but also focuses on the definiteness of node representation. So that nodes could improve the recognition rate while including the dependency information. In order to improve the accuracy of relationship classification, we consider the uncertain information that affects the prediction results, and use the distributinal reinforcement learning method to adaptively adjust the representation of relationship information.

As shown in Fig. 2, we design two types of attention modules to obtain better node feature representation by drawing global contextual information on local features. For sentence $S$ with length $n$, its embedding matrix shape is $A \in \mathbb{R}^{n \times d}$, where $d$ is the dimension of each word (node). We add a convolution layer top of GCN, and use 3 filters to perform the convolution operation to generate a feature map. Then we get a new word vector matrix with the same dimensions. The filter size is 2, 3, and 4, and each has Two filters. This preserve more details without adding additional parameters. Then the new nodes features matrix is input into two parallel attention modules. Finally, the outputs of these two attention modules are aggregated to obtain better node prediction representation. And than through the classification reinforcement module to optimize the relationship information representation and output the optimal relationship classification results.

### B. Position Attention Module

The key to the task of relationship extraction is to discriminant the representation of word features. However, many works [6, 9] indicate that local features information generated by traditional GCN may lose some important information on the text, this lead to the wrong classification of objects. In order to build rich contextual information on local features, we propose a position attention module. Position attention module encodes long-range contextual information into local features, which enhances feature representation ability. As shown in Fig. 3, new features of spatial contextual information are generated through the following three steps. First, generate the position attention matrix, which models the spatial relationship between any two positions of the features. Then, matrix multiplication is performed between the attention matrix and the original feature. Finally, we perform an element-wise sum operation between the multiplied results and the original features to obtain the final representation of the global contextual information. Next, we will detail the steps to adaptively aggregate the spatial contextual information.

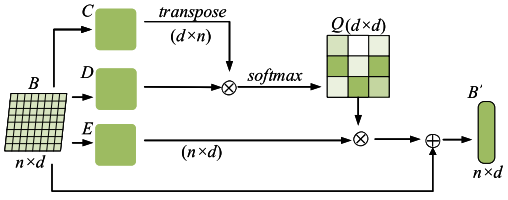


Fig. 3. The position attention module.

First, nodes feature matrix $B \in \mathbb{R}^{n \times d}$ obtained by a convolution layer and then generate three new nodes feature matrix $C$, $D$ and $E$ by convolution operation, where

$\{C, D, E\} \in \mathbb{R}^{n \times d}$. Then, we perform a matrix multiplication operation between the $D$ matrix and the transpose of $C$, and the position attention matrix $Q$ is calculated by using a softmax layer, where $Q \in \mathbb{R}^{d \times d}$,

$$Q_{ij} = \frac{\exp\left(D_j \cdot C_i\right)}{\sum_{i=1}^{d} \exp\left(D_j \cdot C_i\right)}, \tag{1}$$

where $Q_{ij}$ represents the influence of the $j^{th}$ position on the $i^{th}$ position. The more similar the features of the two positions, the greater the correlation between them. Meanwhile, we perform a matrix multiplication operation between $E$ and $Q$. Finally, we multiply the result by a learning factor $\alpha$ with an initial value of 0 and gradually learn to assign more weights [13],

$$B'_j = \alpha \sum_{i=1}^{d} \left(Q_{ij} E_i\right) + B_j. \tag{2}$$

The Equation 2 shows that the final feature of each node is the weighted sum of the features of all nodes and the original features. Therefore, it has global contextual information and selectively aggregates contexts according to the position attention matrix. Similar semantic features promote each other, so as to improve intra-class compactness and semantic consistency.

### C. Relation Attention Module

The dependency information between nodes is very important for relationship classification, and it is also the key to solve multi-hop reasoning. However, many works [3, 4, 5, 6, 7, 8] show that using a pruning strategy and traditional end-to-end model will lose some important dependency information in the text, this could lead to the relationship being misclassified. In order to effectively learn the nodes dependency information, we propose a relation attention module. By emphasizing the feature mapping of interdependence, the representation of semantic features of nodes can be improved. The relation attention module encodes the node dependency information as the relationship feature that to enhance the dependency between nodes. As shown in Fig. 4, the module generates the new feature of node relationship dependency information through three steps. Unlike the position attention module, it directly calculates the relation attention matrix $P \in \mathbb{R}^{n \times n}$. The relation attention matrix is generated according to the dependency between nodes. For example, the relationship between node $i$ and node $j$ is represented by $P_{ij}$ or $P_{ji}$. Initially, the $P$ value of the two associated nodes is 1, and the $P$ value of the nodes not associated is 0. Then, the relationship features are generated by self-attention mechanism. Then, we perform a matrix multiplication on attention matrix and original feature. Finally, an element-wise sum is performed between the multiplied results and the original features to obtain the global dependency between nodes.

Specifically, the relation attention matrix $P \in \mathbb{R}^{n \times n}$ is calculated by a softmax layer,

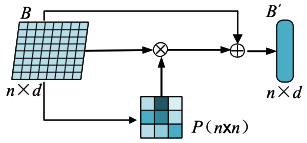


Fig. 4. The relation attention module.

$$P_{ij} = \frac{\exp\left(P_i \cdot P_j\right)}{\sum_{j=1}^{n} \exp\left(P_i \cdot P_j\right)}, \tag{3}$$

therefore, $P_{ij}$ represents the influence of node $j$ on node $i$. The closer the relationship between the two nodes, the greater the impact on this value. Then, we multiply the attention matrix with the original node features. Finally, the result is multiplied by a learning factor $\beta$, then perform an element-wise sum with the original feature to get the final representation $B'$,

$$B'_i = \beta \sum_{j=1}^{n} \left(P_{ij} B_j\right) + B_i, \tag{4}$$

where $\beta$ gradually learns from 0. The final representation contains the dependency information among all nodes, so it can improve the discrimination of node relationship class.

In order to effectively utilize a wide range of contextual information, we aggregate the new features from the two attention modules. Finally, the final node features are obtained by a layer of convolution. The dimension remains the same as the original node feature, so it does not add too many parameters, but effectively enhances the representation of features.

### D. Classification Reinforcement Module

We use distributional reinforcement learning to optimize the representation of relationship features, regard the entities to be classified as states, relationship classification as behaviors, and the deviation between expectation and prediction as rewards, so as to strengthen correct behaviors through rewards. Firstly, the output characteristics of the attention module are represented by the output relationship characteristics of a layer feed-forward neural network (FFNN) [14, 15]. The relationship $r_{ij}$ between entity $h_{ej}$ and entity $h_{ei}$ can be expressed as follows,

$$r_{ij} = \mathrm{FFNN}\left(h_{ei}, h_{ej}, h_{sent}\right), \tag{5}$$

$$h_{sent} = f\left(B'\right), \tag{6}$$

where, $h_{sent}$ means the feature representation of a sentence. The function $f : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^{d}$ converts $n$ vectors into a sentence vector. Then, a probabilistic prediction of the relationship feature of the output is made by $softmax$ function:

$$\mathbb{P}\left(r_{ij} | h_{ei}, h_{ej}, h_{sent}\right) = \mathrm{soft\,max}\left(\mathrm{MLP}\left(r_{ij}\right)\right), \tag{7}$$

where, MLP(.) is a multilayer perceptron. The probability prediction value is transformed into the state value matrix $Q$. Our purpose is to get the optimal expectation value through

iteration, which is obtained from the bellaman optimization formula,

$$Q(h,r) = \mathbb{E}R(h,r) + \gamma \mathbb{E}Q\left(h', r'\right), \tag{8}$$

where, $h$ and $r$ represent the entity to be classified and the corresponding relationship, $Q(h,r)$ represent the cumulative return of $r$ when the action is executed in $h$ state, and $\gamma$ is a penalty factor $(0 \leq \gamma \leq 1)$. Reinforcement learning focuses on the expectation of reward value in the future. The essence of valuation is to predict what hasn't happened, which inevitably involves uncertainty, and the size of uncertainty has a very important impact on decision-making [16] [17]. Therefore, we use distributional reinforcement learning to replace the expected value of the learning return value with the probability distribution of the learning return value, not only to estimate the expected value, but also to estimate the entire distribution function. From the distribution Bellaman operator formula,

$$Z(h,r) = R(h,r) + \gamma Z(h', r'), \tag{9}$$

where, $Z$ is a random variable, which represents the random variable generated by the return after the behavior $r$ is performed in state $h$. The loss can be calculated by cross entropy,

$$\mathcal{L} = \sum_{s \in S} \sum_{i \neq j} \log Z\left(h, r_{ij} | i, j, s\right), \tag{10}$$

where $\mathcal{S}$ denotes a set of sentences and $s$ a single sentence in set $\mathcal{S}$.

## III. Experiments

### A. Data

This section evaluates the proposed DAGCN model on the TACRED and SemEval datasets to facilitate comparisons with other existing models.

As described in [2], the TACRED dataset contains more than 106K mention pairs extracted from the TACKBP Challenge dataset: it defines 41 relationship types and one special "unrelated" class. The mentions in TACRED are typed: the topics are divided into individuals, organizations, and objects which are subdivided into 16 fine-grained types (such as dates and locations). This paper reports the microaveraged F1 scores on the TACRED dataset.

The SemEval 2010 Task 8 dataset is small (1/10 of TACRED) but has been widely used in recent relational extraction work. The dataset defines nine relational types and one special "other" class [18]. Each sample is annotated with the relationship between two given entities. A total of 8,000 samples from the dataset were used for training, and 2,717 samples were used for testing. This paper follows the official task settings and reports the macroaveraged F1 scores.

### B. Setup

To ensure a fair comparison on TACRED datasets, we followed the evaluation scheme used in [2]: the model with the intermediate validation F1 score was selected from five independent runs, and the testing F1 scores are reported. On the SemEval 2010 Task 8 training set, 800 samples were selected as a validation set. The GloVe vector [19] was used to initialize word embedding. Tests on the verification set show that the settings of $L$=5 and $d$=300 achieved the best results. The microaveraged F1 scores on the TACRED dataset and the macroaveraged F1 scores on the SemEval 2010 Task 8 dataset are given in the experimental results.

### C. Results

The models are compared on the TACRED dataset. The models are divided into dependency-based models and sequence-based models. The dependency-based models include logistic regression classifiers (LR) [2], the shortest path LSTM (SDP-LSTM) [20], the tree-structured neural model (Tree-LSTM) [21], GCN and contextualized GCN (C-GCN) [6]. AGGCN and contextualized AGGCN(C-AGGCN). Among the sequence-based models, the position-aware LSTM model (PA-LSTM) [2] with the best effect is selected. Similar to C-GCN [6] and C-AGGCN [9], we extend DAGCN (C-DAGCN) using a bidirectional LSTM network to capture the context representation embedded in the DAGCN layer.

TABLE I
RESULTS ON TACRED DATASETS.

| Model | P | R | F1 |
|---|---|---|---|
| LR [2] | **73.5** | 49.9 | 59.4 |
| SDP-LSTM [20] | 66.3 | 52.7 | 58.7 |
| Tree-LSTM [21] | 66.0 | 59.2 | 62.4 |
| PA-LSTM [2] | 65.7 | 64.5 | 65.1 |
| GCN [6] | 69.8 | 59.0 | 64.0 |
| C-GCN [6] | 69.9 | 63.3 | 66.4 |
| AGGCN [9] | 69.9 | 60.9 | 65.1 |
| C-AGGCN [9] | 71.8 | 66.4 | 69.0 |
| DAGCN(ours) | 70.1 | 63.5 | 66.8 |
| C-DAGCN(ours) | 72.6 | **68.7** | **70.6** |

As shown in Table I, the logical regression classifier (LR) model achieves the highest precision. Because of feature-based methods tend to predict tags with frequency, this high accuracy and low recall may be caused by data imbalance. As shown in Table I, the neural network model is better able to balance the P and R values. Because the models in this paper are similar to GCN, C-GCN, AGGCN and C-AGGCN and the two models have achieved good results, these are subsequently the primary compared models. The results show that DAGCN is superior to the F1 scores of GCN by 2.8 and of AGGCN by 1.7, and the F1 score of C-DAGCN is better than C-GCN by 4.2 and improves on C-AGGCN by 1.6. One possible reason is that the latter two models lack information about word order or disambiguation. The reason for the difference between DAGCN and GCN is obvious because the GCN is based on pruning trees. Therefore, running on a complete tree combined with an attention mechanism allows the model to better distinguish task-related from irrelevant information.

The evaluation results on SemEval are shown in Table II. The C-DAGCN model still achieves the optimal results. It achieves an F1 score 2.1 points higher than that of C-GCN and 1.2 higher than that of C-AGGCN.

TABLE II
RESULTS ON SEMEVAL DATASETS.

| Model | F1 |
|---|---|
| SVM [22] | 82.2 |
| SDP-LSTM [20] | 83.7 |
| SPTree [8] | 84.4 |
| PA-LSTM [2] | 82.7 |
| C-GCN [6] | 84.8 |
| C-AGGCN [9] | 85.7 |
| C-DAGCN(ours) | **86.9** |

Compared with the model without attention module, we visualize the nodes from the semantic association strength and classification effect. In order to make the results clearer, we sample a small number of sentences. Fig. 5(a) and (c) show sentence node association visualizations with the attention module added, while (b) and (d) show the sentence node association visualization without the attention module. We can see that the model with the attention module strengthens the semantic relevance between nodes. A small number of nodes in (b) and (d) are not associated with entities, while the nodes in (a) and (c) are almost all related to one other. From the distances between nodes, we can see that nodes in (a) and (c) represent stronger relevance.

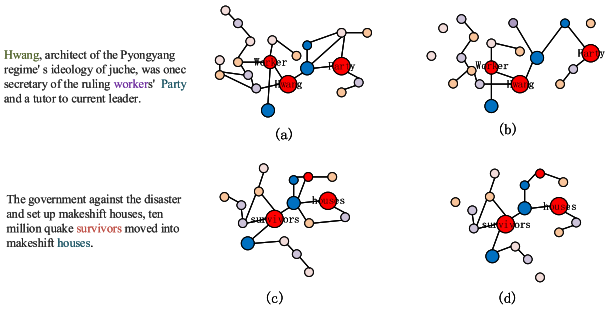


Fig. 5. Visualization of node association. Two sentences, (a) and (c) are shown the results of the model with the added attention module, while (b) and (d) show the results of the model without the attention module.

We classify and visualize entity relationships in a small number of samples, and each sample has at least one entity pair. The different colors represent different relationships, as shown in Fig. 6. The model results without the attention module (see Fig. 6(a)) show that there is some confusion in the entity relationship classification. In contrast, the classification boundaries of the model with the attention module are clearer (see Fig. 6(b)).

Taking the test on semeval data set as an example, we track the learning of relationship classification by adding distributional reinforcement learning. We input 4 time points T, extract one time point every 1000 iterations, set the initial penalty factor $\gamma$ to 0.5, and output the probability distribution diagram of relationship categories, as shown in Fig. 7. In the case of constant penalty factor, with the increase of the iterations after added reinforcement learning, the model can learn the relationship classification distribution more accurately.

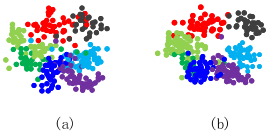


Fig. 6. Visualization of entity relationship classification. (a) is the result of the model with no added attention module model, (b) is the result of the model with the added attention module.

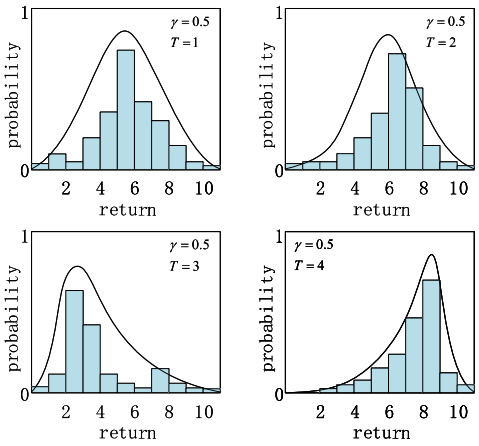


Fig. 7. Distribution results of relation classification learning.

## D. Analysis and Discussion

*1) Ablation Study:* The two main attention modules and classification reinforcement module were selected for the study, and the evaluation results are shown in Table III. From these results, it was found that the results of adding the attention modules resulted in significant improvements. The increased F1 value of 3.5 was achieved by adding the position attention module, and the 3.3 F1 value was achieved by adding the relationship attention module. Together, these two modules increased the F1 value by 4.2. The feedforward neural network has little effect on the results; removing this network decreased the F1 score to 68.1. However, the result increased by 3.1F1 after reinforcement learning was improved. Overall, the two parallel attention modules and classification reinforcement module play important roles and help the GCN learn better aggregated information and generate better representations for a graph structure.

TABLE III
AN ABLATION STUDY FOR C-DAGCN MODEL.

| Model | F1 |
|---|---|
| C-DAGCN | 70.6 |
| $-h_e$, $h_{sent}$, Feedforward (FF) | 68.1 |
| −Classification Reinforcement Layer (CR) | 67.5 |
| −Position Attention Layer (PA) | 67.1 |
| −Relation Attention Layer (RA) | 67.3 |
| −PA, RA | 66.4 |

*2) Performance under Varied Training Data Sizes:* We set up five types of training data (20%, 40%, 60%, 80% and 100%) and evaluated the performances of the C-GCN, C-AGGCN and C-DAGCN models. As shown in Fig. 8, as the training data increase in scale, the performance gap among the three models becomes more obvious. When the size of the dataset reaches 80%, the F1 value of C-DAGCN is close to the maximum value of C-AGGCN. These results show that the training dataset is more effectively utilized by the model proposed in this paper.

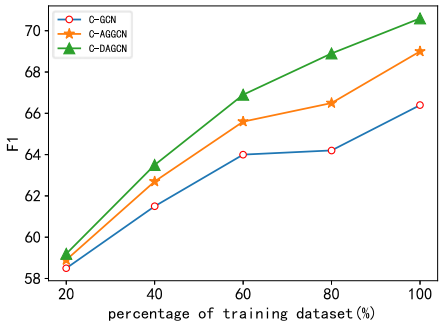


Fig. 8. Comparison of C-AGGCN, C-GCN and C-DAGCN under different training data quantities. The C-GCN results are from [6], and the C-AGGCN are from [9].

*3) Performance under Varied Training Sentence Length:* We set up five sentences with different lengths (<20, [20, 30), [30, 40), [40, 50), ≥50) and evaluated the performances of the C-GCN, C-AGGCN and C-DAGCN models. As shown in Fig. 9, all the models are based on complete trees. C-DAGCN outperforms the other two models at different settings. As the sentence length increases, the dependency graph includes more nodes, and the information acquisition performance decreases. However, the results show that C-DAGCN can better obtain more useful information with a larger dependency graph.

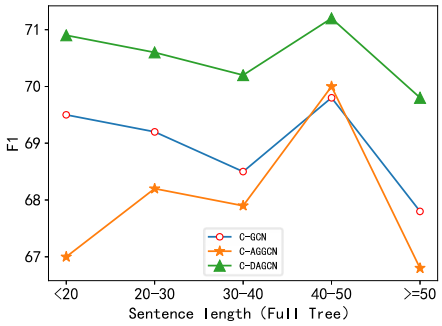


Fig. 9. Comparison of C-AGGCN, C-GCN and C-DAGCN under different sentence lengths. The C-GCN results are from [6], and the C-AGGCN results are from [9].

## IV. Conclusion

In this paper, we introduced a new model that combines dual attention mechanism and distributional reinforcement guide the graph convolutional network for relational extraction. This method uses two parallel attention modules to aggregate global semantic information to enhance the discriminant ability of feature representation. Meanwhile, it combined with GNNs, the problem of multihop relational reasoning and inefficient utilization of dependency tree structure information can be effectively alleviated. The DAGCN model runs on a complete tree and learns to extract useful information from the tree in an end-to-end manner. The two parallel attention modules enable the model to learn more useful information for relational extraction. We believe that the DAGCN model will contribute to improvements in future works. The existing relational extraction model may be helpless faced with lack of information. Thus, unsupervised knowledge extraction will promote progress in future works.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China (Nos. 61966004, 61663004, 61866004, 61762078, 61967002), the Guangxi Natural Science Foundation (Nos. 2019GXNSFDA245018, 2018GXNSFDA281009, 2017GXNSFAA198365), the Guangxi Bagui Scholar Teams for Innovation and Research Project, the Guangxi Talent Highland Project of Big Data Intelligence and Application, Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing.

## References

[1] M. Yu, W. Yin, K. S. Hasan, C. N. dos Santos, B. Xiang, and B. Zhou, "Improved neural relation detection for knowledge base question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 571–581, 2017.

[2] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 35–45, 2017.

[3] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pp. 2335–2344, 2014.

[4] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention cnns," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[5] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W. Yih, "Cross-sentence n-ary relation extraction with graph lstms," *TACL*, vol. 5, pp. 101–115, 2017.

[6] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2205–2215, 2018.

[7] K. Xu, Y. Feng, S. Huang, and D. Zhao, "Semantic relation classification via convolutional neural networks with simple negative sampling," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 536–540, 2015.

[8] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[9] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 241–251, 2019.

[10] H. Zhu, Y. Lin, Z. Liu, J. Fu, T. Chua, and M. Sun, "Graph neural networks with generated parameters for relation extraction," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1331–1339, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[13] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 7354–7363, 2019.

[14] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 1506–1515, 2017.

[15] T. Fu, P. Li, and W. Ma, "Graphrel: Modeling text as relational graphs for joint entity and relation extraction," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1409–1418, 2019.

[16] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 449–458, PMLR, 2017.

[17] W. Dabney, Z. Kurth-Nelson, and N. Uchida, "A distributional code for value in dopamine-based reinforcement learning," *Nature*, pp. https://doi.org/10.1038/s41586–019–1924–6, 2020.

[18] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. W. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 4967–4976, 2017.

[19] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 188–197, 2017.

[20] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pp. 33–38, 2010.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, 2014.

[22] Y. Xu, L. Mou, G. Li, Y. 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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1785–1794, 2015.