Research and Application of Relation Extraction based on Triple Relation Graph Convolutional Networks

Dezhi An, Xuejie Ma, Cao Jiang, Lei Liu and Yanxu Wang
School of Cyber Security, Gansu University of Political Science and Low, Lanzhou, China
Email: maxuejie514@163.com

Abstract. In recent years, data in non-Euclidean spaces is becoming more and more. Traditional methods cannot perform feature extraction on these data. Most of existing methods just extract contextual semantic features from relational instances. Their structural features in corpora are ignored. To solve this problem, the paper proposed a relation extraction method based on triple relation graph convolutional networks (TRGCN). Based on the extraction of semantic features of sentences using convolutional neural networks, this method used the concept of triple relation graphs to represent structural features. In other words, triple relation graphs were formed by considering triples formed by the relation between two entities in one sentence as nodes and triples with common entities and same relations as edges. Finally, multiple-layer graph convolutional networks were used for training. As shown by experimental results, the method proposed in this paper achieved an F1 value of 86.8% on the SemEval 2010Tes8 data set, indicating that it is better than mainstream convolutional neural networks and recurrent neural networks.

Keywords: Information extraction; relation extraction; triple relation graph; convolutional neural network; graph convolutional network

1. Introduction
Information extraction mainly studies how to transform natural texts with unstructured descriptions into structured data. Relation extraction (RE) is a challenging task in the field of information extraction. As the cornerstone of large-scale relational comprehension and application on unstructured texts, it has extensive applications in question answering systems, knowledge graphs and other fields. For example, in the sentence of “The school <e1>master</e1> teaches the lesson with a <e2>stick</e2>.”, the relation between “master” and “stick” is determined as Instrument-Agent (e2, e1) in accordance with the given sentence and the marked entity pairs. The earliest classification model of relation extraction depended on manual features and kernel function construction methods. However, the ability of these methods to capture sentence information is extremely low. The captured information is prone to errors, so the classifier performance of these methods is generally low. Compared with traditional methods with error transmission problems, the present popular methods based on deep learning can avoid such errors because they don’t need to rely on manually constructed features. For example, convolutional neural networks (CNN) and recurrent neural networks (RNN) have made good achievements in both computer vision and natural language processing. However, due to the generation of increasing data, the form of data is becoming more and more diversified. Currently, there are data structures that cannot be solved by CNN and RNN or solved with poor results, namely, graph-structured data. Images or languages used by deep
learning-based methods are all dimensional European spatial data. Such data has an advantage that is quite beneficial to data processing, which is that its structure is very regular and easy to operate. But in real life, in addition to regular European data, there are many irregular non-European data structures, such as graph data structures, or topological structures, as well as social networks, chemical molecular structures, knowledge graphs; even for a natural language, the information it contains may also be a complex tree structure, namely, graph-structured data. The structure of graph data is quite irregular, which does not have the translation invariance law of European data, because the structure around each node in the graph may be quite different. Therefore, for data with this structure, CNN and RNN in traditional deep learning methods do not work. As the scale of graph data continues to expand, the demand for effective processing and analysis of graph data is becoming stronger. Researchers keep trying to apply convolutional neural networks to the processing and analysis of graph data so as to develop a universal network model. Starting from the convolution theorem based on the graph theory, Bruna et al. [2] first proposed a graph convolution network. The graph convolution operation in this model was defined in the spectral space, laying the foundation for subsequent spectral methods in the field of graph convolution; Defferrard et al. [3] introduced Chebyshev polynomials based on traditional convolutional neural networks, thus obtaining free parameters to approximate the filters in the spectral method to learn local, fixed and combined features on the graph; Kipf et al. [4] parameterized the convolution kernel in the spectral method, and simplified the traditional graph convolution networks, which greatly improved the training efficiency and accuracy of the model. Therefore, for relation extraction performed on graph-structured data, this paper proposed a relation extraction method using graph convolutional networks. During the capture of the semantic features of sentences, a triple relation graph was proposed to extract the structural features of the corpora to assist relation extraction. Eventually, the proposed model was tested on the SemEval2010-Task8 data set, which achieved better results than existing models.

2. Relevant Work

Relation extraction is an essential research direction in the field of information extraction. Nowadays, there are a large number of data sets with labeled entities among abundant data sets. Normally, relation extraction is regarded as a classification problem. In other words, the entity pair corresponding to the relation has been marked in the data set instance, and then the classifier is used to determine the relation type of the relation instance. Traditional relation extraction methods mainly include methods based on features and methods based on kernel functions. The methods can indeed achieve certain results, but depends on the feature set selected by the methods or the designed kernel functions designed. This characteristic makes the methods prone to artificial errors and error propagation, which greatly limits the performance of the relation extraction model. In recent years, with the development of deep learning, the application of deep learning to the field of natural language has become increasingly mature. Relation extraction has also gradually transitioned from traditional methods to deep learning methods. The development process of these methods will be introduced in detail below. Methods based on features describe the relation between entities by focusing on the extraction of important features in the text, such as entity type, dependency tree, and word block feature, which will be organized vector forms. Then, machine learning algorithms (such as support vector machine, maximum entropy) will be used to classify the extracted relation features. Such methods mainly depend on the design of features and the precision of natural language processing tools used (such as named entity recognition, part-of-speech tagging, phrase extraction). Therefore, such methods need a great deal of time, which are not only time-consuming but also prone to the accumulation and propagation of errors, thereby affecting the final classification performance. Methods based on kernel functions obtain the structural characteristics of objects and establish a classification model by calculating the similarity of objects in a high-dimensional space through the designed kernel function. However, such methods rely on the design of the kernel function, which are also affected by the loss of natural language processing tools, so the generality is not poor. Methods based on neural networks can automatically learn text features without manual construction of features. Therefore, they can
avoid the losses caused by traditional methods and improve the accuracy of the classifier. There are two main methods of neural networks, one is based on convolutional neural networks, and the other is based on recurrent neural networks. Among them, the methods based on convolutional neural networks are mainly as follows. Zeng et al. proposed the use of convolutional neural networks in the modeling of relation instances, and introduced position vectors [5]; Nguyen et al. Characterized semantic relations by designing convolution kernels of different scales [6]; Santos et al. proposed the use of hinge loss functions for the replacement of the original cross-entropy to better distinguish different types of relation instances [6]; Wang et al. tried to use a multi-layer attention mechanism in convolutional networks to highlight the contribution of sentence components to relation labels [8]. The methods based on recurrent neural networks are mainly as follows. Zhang et al. proposed to use RNN to replace CNN for the modeling of relation instances. The position vectors were replaced by simple position labels [9]; Li et al. used recurrent neural networks to model different syntactic structures [10]; Miwa et al. used Tree-LSTM to model relational instances and considered types of different syntactic structures [11]; Zhou et al. applied bidirectional recurrent neural networks and attention mechanism to relation extraction tasks [12]; Xiao et al. cut long sentences and used two-layer RNN as well as attention mechanism to encode them separately [13]. For the model mentioned above, an end-to-end model was constructed and sentence instances were directly input into the model. This method also achieved good results. However, most present relation extraction models regard target entity pairs as independent objects [14][15]. They only extract context-based semantic features from relation instances through neural networks while ignore the relation of triples in each instance at the corpus level. Considering the above-mentioned problems and deficiencies, this paper studied the triples contained in all instances. If there were entities the same as those in the target entity triples, it was considered that it can assist the extraction of relations between target entities. Specifically, an entity in an instance does not only appear in the current instance, or only has a relation with another entity in the current instance. When the data set is large enough and the data corpora is sufficient, the entity in the current instance not only has a relation with other entities in the current instance, but may also have potential relations with entities in other instances. A triple relation graph can be obtained by extracting the relations between all entities in the entire corpus. To effectively capture the relations between entities, the text semantic feature is constructed to support the relation extraction task. In other words, the feature extraction method of convolutional neural networks is used. The high-frequency word 0/1 of the sentence are used as the feature representation of the sentence. Facts proved that the method above can indeed improve the relation extraction precision of graph convolutional networks.

3. Model Description and Implementation
This chapter mainly introduces the relation extraction model of graph convolutional networks based on triple relation graphs. The model framework is shown in figure 1:
First of all, this paper obtained the semantic features of each sentence through convolution, and then collected the high-frequency words in all sentences. Next, it determined in turn whether the words in each sentence were included in the high-frequency words. 1 indicated yes and 0 indicated no. In this way, the 0/1 feature vectors of the sentences can be obtained. The two parts were added together as the feature vectors of the sentences. Then a triple relation graph was constructed. The triple of each sentence was considered as one point. If the two triples had a common entity and the relation was the same, then it was assumed that there was an edge between the two points. The number of sentences in the data set was the number of points. At last, a triple relation graph can be obtained as the initial feature of the graph convolutional network (GCN). After that, convolution and nonlinear transformation operations were performed on sentence features through a two-layer GCN to obtain the final representation. The log_softmax function was used to normalize it to obtain the final classification result.

3.1. Vector Presentation Layer
Currently, mainstream neural network relation extraction methods usually use word vectors to embed the contextual features of relation instances. Compared with traditional feature extraction methods based on lexical or syntactic analyzers, such representation methods can dig up the deep semantic information in relation instances more sufficient. At the same time, the extensibility for various corpora is stronger. Therefore, in the model described in this paper, the word vector in the experiment used the pre-trained Glove word vector to embed the word vector in the corpora. In addition, considering that the position of the entity in the sentence can reflect the position information between each word and the entity pair in the sentence, the position feature proposed by Zeng et al. [5] was introduced as well. The position embedding representation was obtained by randomly initializing the position embedding matrix. The position embedding vectors of the word relative to the entity were $P^1_l$ and $P^2_l$. Therefore, the word vector of the word can be expressed as $x_i = [w_i; P^1_i; P^2_i]$. The vector representation of the relation part of the sentence in this experiment was represented by one-hot vector. The data set used in this experiment had 19 kinds of relations, so each relation was represented by a 19-bit vector. Only one of the 19 bits was valid to express this relation.
3.2. Sentence Feature Presentation Layer

After inputting the word vector obtained from the vector presentation layer to the convolutional neural networks layer, the information in the relation instance is usually contained in the local area of the sentence. For the extraction of local area information, the convolutional neural network method is used to extract the local information of the sentence. Let the input sentence matrix as \( C \), and the dimension of each word is \( d \) (including the \( d^w \)-dimensional word vector representation and the \( d^p \)-dimensional position vector). To obtain the feature representation of the input sentence, the convolution model uses a filter with an initial window size of \( K \) for the convolution operation, and its width is consistent with the dimension of the word vector. The convolution operation is shown in the following equation (1).

\[
\begin{align*}
   c_i^k &= \sigma \left( \sum (C[i; i + k] \cdot H_k) + b \right) \\
\end{align*}
\]  

(1)

Among them, “\( \cdot \)” represents the dot product, \( \sigma \) represents the sigmoid activation function, represents the word vector sequence from \( i \) to \( i+k \), is a convolution kernel with a width of \( k \). Finally, the feature matrix fused with local information is obtained \([c_1, c_2, \ldots, c_n]\). These features can be used to represent part of sentence features. For the representation of rest sentence features, the words in all instances in the data set are collected first and then sort from high frequency to low frequency. The first 1000 words are taken as high frequency words. The 0/1 vector is constructed by verifying whether the words in the instance are included in the high frequency words. As a result, a 0/1 matrix of the sentence features can be obtained. The features of sentences can be represented with the combination of the above-mentioned semantic feature extraction.

3.3. Graph Convolutional Network

Graph convolutional network (GCN) is a simple and effective convolutional neural network based on graph-structured data. Since it can effectively capture the dependence between data through the information transfer between graph data nodes, it has been widely used to process data with abundant relations between objects and interdependent relations. GCN acts directly on the graph [16]. The input of the network includes the structure of the graph and the characteristic representation of the nodes in the graph. For each node in the graph, GCN obtains the feature representation vector of the node by fusing and summarizing the properties of other nodes around.

This model architecture was designed and proposed by kipf and Welling [4] in 2016, which can directly calculate graph-structured data. Specifically, for given a graph \( G=(V, E) \), \( V \) is a vertex set containing \( N \) nodes, and \( E \) is an edge set including self-loop edges (namely, each vertex is connected to itself). Kipf and Welling [4] used as a feature matrix, where the feature dimension of each node is \( d_k \), and the \( i \)-th row vector in the matrix represents the feature of the \( i \)-th node \( \nu_i \). The element in the adjacency matrix \( A \in \mathbb{R}^{N \times N} \) indicates that whether there is a connection between the \( i \)-th node and the \( j \)-th node in the graph. Generally, if there is a connection between two nodes, the value of is 1, otherwise it is 0 [17]. In practical applications, the performance of multi-layer GCN is often better than that of single-layer GCN because it can integrate a wider range of node information. Specifically, the \( l \)-th layer encodes the output of the \( l-1 \)-th layer, and the calculation is shown in the following equation (2) [17].

\[
L^l = f \left( A L^{l-1} W_l + b_l \right) 
\]  

(2)

Among them, is a parameter matrix that can be learned, is the bias, and is the activation function, which can perform nonlinear transformation on the output. indicates the layer of GCN, the output of the GCN of the 0-th layer is the node feature matrix \( X \), that is \( L^0 = X \). Graph convolutional networks perform convolution operations on the feature matrix by sharing parameter \( W_l \). Because of the sharing of local parameters, GCN can prevent overfitting to a certain extent. During the processing of the text, a GCN represented by the sentence feature is constructed as the node. Then, the semantic feature vector of the node A can be updated through the neighbor nodes in the receptive field of the node A to obtain the feature representation containing the semantic information of neighbor nodes.
3.4. Graph Convolutional Layer based on Triple Relation Graphs

The graph that GCN acts on is a triple relation graph described in this paper. The triples in the instance are taken as the nodes of the graph. If the two triples have one common entity and the relation is the same, then it is regarded as the two triples are related to each other. In the figure, it is indicated that there is an edge between these two points, and the triples that have the same entity as these two triples are very likely to have the same relation with these two triples. A triple relation graph can be obtained after traversing all the instances in turn, which can also be regarded as an entity relation feature. The model is shown in figure 2 below:

![Figure 2. Triple Relation Graph.](image)

After the text semantic feature extraction and the entity association feature extraction, the TRGCN model uses the softmax classifier to classify the input relation instances based on the text feature representation and the triple relation graph features. The classification result is output by the output layer of the graph convolutional network. The specific form is shown in the following equation (3):

$$z = f(X, A) = \text{softmax}(\hat{A} \text{ReLU}(\hat{A}XW^0)W^1)$$  \(\#(3)\)

Among them, is the weight matrix from the input layer to the hidden layer. The hidden layer is mapped by H features, , is the weight matrix from the hidden layer to the output layer, is the total number of preset relation types, which is the model hyper-parameter. Then, the TRGCN model uses the output of the softmax classifier to predict the relation type between the target entity pairs in the input relation instance, as shown in the following equation (4):

$$\hat{y} = \arg \max_y z$$  \(\#(4)\)

Next, after the evaluation of the predicted label of the input relation instance, the softmax classifier uses the cross-entropy loss function, as shown in the following equation (5):

$$\mathcal{L} = -\sum_i \sum_j \text{label}_{ij} \log(z_{ij} | S_i, \theta)$$  \(\#(5)\)

Among them, indicates the set of all parameters in the model, indicates that the predicted label of the sample i is j, is a one-hot vector about the sample , When the true label is j, is 1, otherwise it is 0. In the TRGCN model, the Adam optimization algorithm is used to solve the parameters, and the network parameters are initialized using the uniform_initialization method.

4. Experimental Results and Analysis

To evaluate the effectiveness of the proposed TRGCN model on the relation extraction task, this paper conducted experiments on the reference data set of relation extraction. Section 3.1 introduces the experimental data set, the division and evaluation indicators. Section 3.2 introduces specific evaluation indicators and parameter settings. Then, the remaining sections elaborate on the experimental performance of the model and its comparison with other methods.

4.1. Experimental Data and Evaluation Indicators

For the relation extraction experiment, this paper used the relation extraction data set SemEval-2010 Task8 of the semantic evaluation conference SemEval, containing 8000 training samples and 2717 test
samples. These samples were divided into 9 types of relations and one other relation type. The relation type and the expected distribution are shown in table 1 below.

On the data set, this paper used the official evaluation indicator macro average (Macro) F1 value evaluation model. Macro first calculates the F1 value for each class, and then averages the arithmetic values for all classes. Table 2 shows the confusion matrix of the relation classification results. Before calculating the F1 index value, the precision rate P, the recall rate R and the correct rate ACC are obtained in accordance with the confusion matrix (shown in table 2 below). The calculations are shown in the following equations (6), (7), (8).

\[ P = \frac{TP}{TP + FP} \times 100\% \]  
\[ R = \frac{TP}{TP + FN} \times 100\% \]  
\[ ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \]

The F1 value is defined as the harmonic average of the precision and the recall, as shown in the following equation (9):

\[ F_1 = \frac{2PR}{P + R} \times 100\% \]

**Table 1.** SemEval-2010 Task8 Data Set Relation Type and Corpus Distribution.

| Relation Type       | Description                | Training Set |          | Tester |          |
|---------------------|----------------------------|--------------|----------|--------|----------|
|                     |                            | Number of Instances | Proportion/% | Number of Instances | Proportion/% |
| Other               | Other                      | 1410         | 17.63%   | 454    | 16.71%   |
| Cause-Effect        | Cause and Effect           | 1003         | 12.54%   | 328    | 12.07%   |
| Component-Whole     | Component and Whole        | 941          | 11.76%   | 312    | 11.48%   |
| Entity-Destination  | Entity and Destination     | 845          | 10.56%   | 292    | 10.75%   |
| Product-Producer    | Product and Producer       | 717          | 8.96%    | 261    | 9.61%    |
| Entity-Origin       | Entity and Origin          | 716          | 8.95%    | 258    | 9.50%    |
| Member-Collection   | Member and Collection      | 690          | 8.63%    | 233    | 8.58%    |
| Message-Topic       | Message and Topic          | 634          | 7.92%    | 231    | 8.50%    |
| Content-Container   | Content and Container      | 540          | 6.75%    | 192    | 7.07%    |
| Instrument-Agency   | Instrument and Agency      | 504          | 6.30%    | 156    | 5.74%    |
Table 2. Confusion Matrix of Classification Results.

| Real Situation | Prediction Result |
|----------------|-------------------|
| True           | False             |
| True           | TP (True Positive) |
| False          | FP (False Positive) |
| False          | FN (False Negative) |

4.2. Parameter Setting

The word vector used in this experiment was a pre-trained Glove word vector with a dimension of 300. In the hyper-parameter setting of the model, the optimal hyper-parameters were obtained after multiple experiments with different hyper-parameters. The experimental range and final settings of the model specific parameters are shown in table 3.

Table 3. Test Parameters.

| Name                      | Variable   | Value |
|---------------------------|------------|-------|
| Batch Quantity            | Batch_size | 256   |
| Learning Rate             | Learning_rate | 0.01 |
| Hidden Layer Parameter    | Hidden_size | 32    |
| Word Vector Dimension     | Word_dim   | 300   |
| Position Vector Dimension | Position_dim | 70   |
| Number of Network Layers  | L          | 2     |
| Dropout Rate              | Dropout    | 0.5   |

In this paper, the precision P (Precision), the accuracy ACC (Accuracy) and the F1 value were used as evaluation indicators to measure the performance of the model. First, the values of P, ACC, and F1 of each category were obtained through the confusion matrix. Then, Macro-average was used to obtained the arithmetic average of all categories as an indicator of the overall performance.

4.3. Experimental Results and Analysis

This section introduces the experimental results of the TRGCN model proposed in this paper on the SemEval 2010 data set, and conducted a comprehensive analysis of the experimental results to prove the effectiveness of the TRGCN model proposed. The TRGCN model proposed is compared with other mainstream relation extraction models on the SemEval 2010 data set. The comparison results are as shown in table 4.

Table 4. Experimental Result Comparison of This Model n SemEval Data Set.

| Type                      | Model      | Value of F1 |
|---------------------------|------------|-------------|
| Feature-based             | SVM        | 82.2        |
| End-to-End Methods        | Multi-CNN  | 82.8        |
|                           | CR-CNN     | 84.1        |
|                           | Bi-GRU     | 83.3        |
|                           | BiLSTM+ATT | 84.0        |
| Dependency Methods        | SDP-LSTM   | 83.7        |
|                           | DRNNs      | 85.8        |
|                           | C-GCN      | 84.8        |
| Our Methods               | TRGCN      | 86.8        |

It can be seen from table 4 that methods based on convolutional neural networks and recurrent neural networks do not use external natural language processing tools. They only extract features from instances for relation extraction, which can also achieve good results. It indicates that the target entity
relation instance contains rich semantic and structural features, which does not rely on external resource. The mining of information from the relation instance can also achieve a good relation extraction effect. In addition, in the comparison method based on the dependency analyzer, the DRNNs model proposed by Xu et al. [18] used the Stanford parser to analyze the relation instance to obtain the dependency relation, which then was input into a deep recurrent neural network for feature extraction. It can achieve good results, indicating that a deeper model can obtain more effective semantic and structural information in relational instances. It can be seen from the above that the F1 value of the method using external natural language processing tools or resources is higher than the method that simply extracts features from the relation instance end-to-end, which proves that external resources can effectively improve the effectiveness of the model. However, the TRGCN model in this paper is better than the above-mentioned methods that use external information and deeper models without using any external resources. This further verifies the starting point of this paper, that is, the corpus itself contains rich semantics and structural information, which also proves that the triple relation graph can effectively improve the relation extraction effect.

5. Conclusion and Prospects
The basic idea of the relation extraction method proposed in this paper is to mine contextual semantic features and structural features required by the relation extraction task from the data set without relying on external resources. Then, it uses the convolutional neural network to obtain sentence features, which are used as the semantic text feature of the sentence with the combination of the 0/1 feature vector of high-frequency words in the sentence. The constructed triple relation graph is used as the structural feature of the data and works together on the relation extraction of graph convolutional networks. Their effects on the model are verified on the public data set. Although the method based on graph convolution has achieved the latest results, this is just a preliminary attempt of the basic idea. There are still many shortcomings, which need further research and exploration.

Acknowledgement
This project was supported by “Science and Technology Innovation Team of Gansu Provincial Education Department (Grant No.2017C-16)”, “Research on Online Public Opinion Situation under Big Data Environment (2018 University-level scientific research and innovation Team of Gansu University of Political Science and Law)” and Collaborative Innovation of Social Network Public Opinion Information (Major Projects of Gansu University of Political Science and Law).

References
[1] Huang H Q, Yu J, Liao X, et al. 2019 Review on knowledge graphs Computer Systems & Applications 28(6): 1-12.
[2] Bruna J, Zaremba W, Szlam A, et al. 2013 Spectral networks and locally connected networks on graphs ArXiv:1312.6203.
[3] Defferrard M, Bresson X, Uanderghenst P 2016 Convolutional neural networks on graphs with fast localized spectral filtering Proceeding of Advances in Neural Information Processing Systems: 3844-3852.
[4] Kipf T N, Welling M 2016 Semi-supervised classification with graph convolutional networks ArXiv: 1609.02907.
[5] Zeng D, Liu K, Lai S, et al. 2014 Relation classification via convolutional deep neural network Proceedings of the 25th International Conference on Computational Linguistics COLINU 2014: 2335-2344.
[6] Nguyen T H, Grishman R 2015 Relation extraction; Perspective from convolutional neural networks Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing:39-48.
[7] Santos C N, Xiang B, Zhou B 2015 Classifying relations by ranking with convolutional neural networks arXiv:1504.06580.
[8] Wang L L, Cao G, Ucrard do Melo, et al. 2016 Relation classification via multi-level attention CNNs. Proceedings of the 54\textsuperscript{th} Annual Meeting of the Association for Computational Linguistics: 1298-1307.

[9] Zhang D, Wang D 2015 Relation classification via recurrent neural network. arXiv:1508.01006.

[10] Li J, Luong M T, Jurafsky D, et al. 2015 When are tree structures necessary for deep learning of representations. arXiv: 1503.00185.

[11] Miwa M, Bansal M 2016 End-to-end relation extraction using lstms on sequences and tree structures. arXiv: 1601.00770.

[12] Zhou P, Shi W, Tian J, et al. 2016 Attention-based bidirectional long short-term memory networks for relation classification. Proceedings of the 54\textsuperscript{th} Annual Meeting of the Association for Computational Linguistics: 207-212.

[13] Xiao M, Liu C 2016 Semantic relation classification via hierarchical recurrent neural network with attention. Proceedings of COLINU:1254-1263.

[14] Miwa M, Bansal M 2016 End-to-end relation extraction using LSTMs on sequences and tree structures. Proceedings of the 54\textsuperscript{th} Annual Meeting of the Association for Computational Linguistics: 1105-1116.

[15] Nguyen T H, Grishman R 2015 Relation extraction: Perspective from convolutional neural networks. Proceedings of the 1\textsuperscript{st} Workshop on Vector Space Modeling for Natural Language Processing: 39-48.

[16] Duvenaud D, Maclaurin D, Aguilera-Iparraguirre Jorge, et al. 2015 Convolutional networks on graphs for learning molecular fingerprints. Proceedings of Advances in Neural Information Processing Systems (NIPS): 2224-2232.

[17] Marcheggiani D, Titov I 2017 Encoding sentences with graph convolutional networks for semantic role labeling. arXiv: 1703.04826.

[18] Xu Y, Jia R, Mou L, et al. 2016 Improved relation classification by deep recurrent neural networks with data augmentation. arXiv: 1601.03651.