GRAPH CONVOLUTIONAL NETWORKS & ADVERSARIAL TRAINING FOR JOINT EXTRACTION OF ENTITY AND RELATION

Xiaolong QU[1], Yang ZHANG[2], Ziwei TIAN[3], Yuxun LI[4], Dongmei LI[5]*, Xiaoping ZHANG[6]*

Entity recognition and relation extraction are the core tasks in information extraction. Currently, supervised deep learning extraction methods are mainly divided into two categories: pipeline and joint entity-relation extraction. The pipeline method has problem of exposure bias, information redundancy, error accumulation and interaction missing. To solve the problems, researchers proposed joint entity-relation extraction method. However, the joint entity-relation extraction method based on sequence annotation does not effectively process entity overlapping, and relation overlapping. Therefore, we propose a joint extraction model GcnJere based on graph convolutional neural network to solve existing problems in the pipeline method and further improve the processing effect of entity overlapping and relation overlapping. Furthermore, we combine the advantages of adversarial training and propose GcnJereAT to improve the generalization ability and robustness of GcnJere. Finally, the performance of the proposed two models is verified in the public benchmark dataset. The experimental results indicate that the computational performance of the two models is superior to the comparison models.

Keywords: graph convolutional network, adversarial training, entity recognition, relation extraction

1. Introduction

Relation extraction, the critical-tasks in the field of information extraction, aims to convert unstructured text to structured data and promoting the automatic construction of the knowledge base. Existing supervised deep learning relation extraction method is mainly classified into two kinds: pipeline and joint extraction.

[1] M.S., School of Information Science and Technology, Beijing Forestry University, China, e-mail: quxiaolong@bjfu.edu.cn

[2] Eng., School of Information Science and Technology, Beijing Forestry University, China, e-mail: 17712433327@163.com

[3] M.S., School of Information Science and Technology, Beijing Forestry University, China, e-mail: tian15501233099@163.com

[4] M.S., School of Information Science and Technology, Beijing Forestry University, China, e-mail: Lyx0421@bjfu.edu.cn

[5] Prof., School of Information Science and Technology, Beijing Forestry University, China, e-mail: lidongmei@bjfu.edu.cn, * Corresponding author

[6] Prof., National Data Center of Traditional Chinese Medicine, China Academy of Chinese Medical Sciences, China, e-mail: xiao_ping_zhang@139.com, * Corresponding author

For the sentence "*Beijing is the capital of China.*", the pipeline method performs two processes, named entity recognition and relationship extraction, in sequence. First, the entities "*Beijing*" and "*China*" are recognized, followed by the relationship "*capital of*" between them is recognized and then the subsequent relationship extraction process is exited. This method inevitably suffers from propagation errors, redundancy and missing information. Although the sequence labeling-based joint extraction method integrates the two processes in the same framework, it ignores the problem of overlapping relational triples, which can recognize the relationship "*capital of*" but cannot further extract the relationship "*located in*". In contrast, the joint extraction method based on graph convolutional networks (GCN) aggregates feature information between words with dependencies in a sentence, effectively improving the defects of existing methods and improving the accuracy of joint entity relationship extraction [1,2].

In addition, the existing neural network models have common defects of adding a slight disturbance to them, which can easily change the final results, resulting in high confidence in the wrong decision. Therefore, we introduced adversarial training (AT) [3,4] to strengthen the robustness of models. AT is a training method that introduces noise and disturbance. Compared with the original input, the added disturbance is slight, but it can make the model predict error. For example, x in the triple (*Beijing, located in,* x) must be location information.

In summary, we apply GCN and AT to enhance joint entity-relation extraction method and propose two models: GcnJere and GcnJereAT, which can alleviate the problem of overlapping entities and overlapping relationships to a certain extent. After Bi-Directional Graph Convolutional Networks (Bi-GCN) preliminarily identified entities and relationships, we construct a dense connectivity layer of graph structure with the relationship type as the dimension for capturing local and non-local information of entities and their relationships, and finally integrate the captured information for entity and relationship prediction, which further enhanced the prediction results. We use AT to avoid model overfitting, improve model robustness, confidence, and generalization. In addition, we use the public benchmark dataset to estimate the performance of GcnJere and GcnJereAT.

2. Related Works

There are three kinds of joint extraction methods: parameter sharing, sequence labeling, and graph. Li [5] mainly used the structured system of complex feature engineering to achieve the joint entity and relation extraction. Zheng [6] used LSTM to cope with the question of long-term dependence on labels, solved the information redundancy problem in the pipeline method by using a new annotation mechanism. To address the problem of overlapping of entity and entity pair, Bekoulis et al. [7] regarded the relation between entities as a multi-headed selection problem, which identifies multi-relations of each entity pair as much as possible. According to the degree of overlapping of entity in sentences, Zeng et al. [8] classified relations into three types: normal, entity overlapping, and relation overlapping, and put forward a joint extraction method based on replication mechanism. With the continuous improvement of GCN, it is gradually applied to natural language processing (NLP). Wang et al. [9] firstly regarded the joint extraction as a directed graph problem. Sun et al. [1] used GCN to derive the type of relation between entities, where the detection of entity range and the derivation of relation between entities were carried out simultaneously. Hong et al. [2] combined the advantages of GCN and Bi-LSTM to extract the long-distance entity pairs. Yan et al. [10] used a partitioned filtering network to correctly model the bi-directional interaction between entity and relation extraction.

However, the existing neural network models have some defects. Szegedy et al. [11] found that adding a few slight disturbances to the deep learning network model may lead to high confidence decision-making errors in the model. Goodfellow et al. [12] proposed confrontation samples and AT. Considering that AT can effectively avoid the overfitting phenomenon of the deep neural network during the learning process, researchers gradually applied it to NLP field. Miyato et al. [14] implemented text categorization using adversarial samples. To extract relations. Bekoulis et al. [3] applied AT to joint extraction and verified it with four data sets. And Yasunaga et al. [14] applied AT to improve the overall precision of pos-labeling and the performance of sequence-labeling model. Experimental results indicated that the F1 value has effectively improved 0.4%~2%.

3. GcnJereAT

The framework of GcnJereAT is shown in Figure 1. It is noteworthy that the framework diagram of GcnJere is not given in this paper, because the framework of the two models is consistent. The difference is that GcnJere does not have disturbance data for adversarial training when inputting.

The GcnJereAT consists of the following main steps: 1) The original text is pre-trained to generate high-quality word embeddings as input; 2) GcnJereAT use Bi-directional LSTM(Bi-LSTM) to generate initial features for pre-trained data; 3) GcnJereAT performs preliminary training of the model using Bi-GCN on the pruned dependency tree and the generated adjacency matrix to generate entity boundaries and preliminary relation types; 4) Based on the relation types, GcnJereAT rebuild the graph to generate a Densely connection layer that captures relations between entities more fully; 5) GcnJereAT output the final result from the Combined output layer.

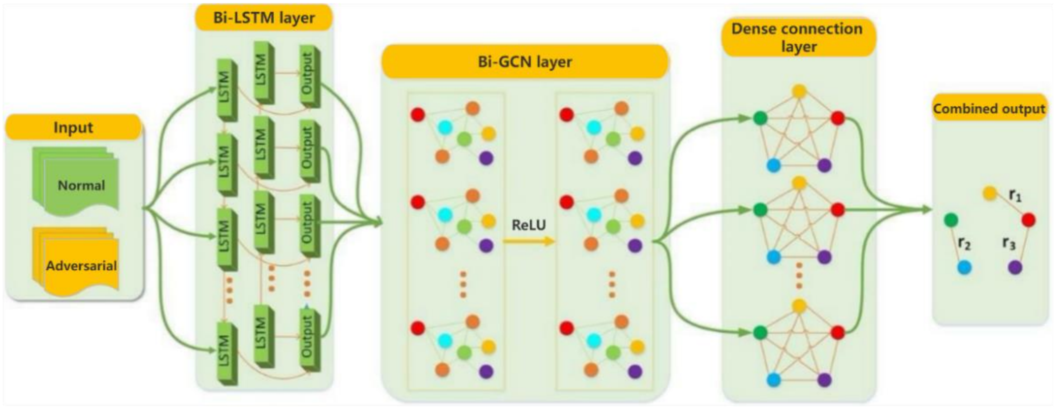

Fig. 1. Framework of GcnJereAT

3.1 Input layer

The existing word embedding methods mostly used word sequence information rather than syntactic information. For the purpose of enhancing the quality of word embeddings, before inputting data into the model, we use the SynGCN method proposed by Microsoft on ACL2019 to train the text in advance [15]. SynGCN uses GCN to learn word embeddings by using context dependencies. For words in the vocabulary, SynGCN learns its representation by predicting words using dependent contexts based on GCN encoding.

3.2 Bi-LSTM layer

Previous studies have shown that the frequency of sentences with length less than 30 words in the text was only about 50%, while GCN cannot effectively obtain the long distance between words within a sentence. Therefore, we apply Bi-LSTM to extract the sequence information. For example, sentence $S = w_1, w_2, \dots w_i, \dots w$, we use w_i and $pos_{i.}$ to calculate the initial features as follows:

$$h_l^0 = w_i \oplus pos_i \qquad (1)$$

where h_l^0 represents the initial feature of $word_i$, w_i and pos_i represent word embeddings and position embedding, respectively.

3.3 Bi-GCN layer

Since the input text is based on the sentence sequence, there is no internal graph structure information. To solve this problem, we build a dependency tree by parsing the input sentences. The dependency tree is used as the adjacency matrix of sentences, and GCN is used to extract partial dependency features. In addition, to further consider the lexical features of input and output, we improve the undirected GCN as Bi-GCN to process the input and output features as the final lexical feature in this paper.

The model uses Stanford CoreNLP for syntactic analysis to obtain long-distance syntactic relations. Relation extraction usually requires abundant syntactic features to improve model performance, while previous studies focused on using the dependency path with the shortest distance between two entities to obtain relation information However, the dependency path with shortest distance may ignore serval important information. With the help of a syntactic dependency tree, abundant sentence structure information can be obtained. Usually, the most useful information between two entities in one sentence is contained in a subtree with the nearest common ancestor of the two entities as the root node, which provides some ideas for reducing redundant information. Removing irrelevant information from the LCA subtree can improve the performance of the model. Moreover, since the size of the tree affects the size of the fully connected graph in GCN, we use a pruning strategy to further speed up the training process of the model. The core of the pruning strategy is finding the dependency path of the dependency tree, specifically obtaining the word vector whose dependency distance path is K in LCA subtree. Based on the experience of previous literature, we use the pruning strategy of LCA subtree with distance-dependent path K=1 to reduce the redundant information further and speed up the training of the model.

In this paper, we convert each pruned dependency tree into the corresponding adjacency matrix A, and use GCN to establish the adjacency matrix of the dependency tree. $S = [w_1, w_2, \dots w_i, \dots w_n]$ note the input sentence, where i denotes the position in the sentence and w_i denotes the pre-trained d-dimensional word vector. We use an adjacency matrix to represent the graph structure of the Bi-GCN layer, with $G = \{V, E\}$ as a directed graph, where V denotes the set of nodes and E denotes the set of directed edges. The dependency analysis diagram of the model is represented by the adjacency matrix A_{ij} of $n \times n$, where $A_{ij} = 1$ indicates that there is a dependency relation between node i and j. The weights of different values can distinguish different dependencies. In order to avoid large differences in the results of node representation, we use formula (2) to improve the representation effect of sentences in this paper.

$$h_i^l = \sigma\left(\sum_{j=1}^{n}\left(\frac{\bar{A}W^l h_j^{l-1}}{d_i} + b^l\right)\right) \tag{2}$$

where $\bar{A} = A + 1$, and I represents the unit matrix of $n \times n$, and d_i represents the degree of node i in the graph.

Since the original GCN is an undirected graph, for purpose of comprehensively considering the word characteristics of input and output, we adopt the Bi-GCN method to meet the demand for direct representation. The formulas for Bi-GCN are shown below:

$$h_i^l = \sigma\left(\sum_{j=1}^{n}\left(\frac{\overline{A}_{ij}\overrightarrow{W}^l h_j^{l-1}}{d_i} + \vec{b}^l\right)\right) \oplus \sigma\left(\sum_{j=1}^{n}\left(\frac{\overline{A}_{ij}\overleftarrow{W}^l h_j^{l-1}}{d_i} + \overleftarrow{b}^l\right)\right) \quad (3)$$

where h_i^l represents the hidden feature of word embedding w_i at the L layer. $\overrightarrow{W}$ and $\overleftarrow{W}$ represent the weight matrix of the output input. $\vec{b}$ and $\overleftarrow{b}$ represent the bias vector of the input-output, respectively.

At this stage, we use the entity labeling method of BIESO (Begin, Inside, End, Single, Other) for encoding in this paper. Using the sequence information extracted by Bi-LSTM and the dependency structure characteristics of Bi-GCN coding, the relations between all word entities and word pairs are preliminarily predicted.

3.4 Densely connection layer

Since the Bi-GCN layer does not take the relation between entities and relations into account, for the purpose of considering the impact between entities and relations and the local or non-local information between all the words in the text, this stage proposes to build a completely weighted relation graph for each relation r based on the relation label r as the dimension to train on the Bi-GCN to capture more structural information, and then capture rich side information. Finally, the final prediction of the relation between entities is achieved by integrating the obtained information. The prediction of the relation classification at this stage is more reliable than that of the Bi-GCN layer. In addition, the overlap of entities and relationships can be solved at this stage.

Bi-GCN is continued to be used on each relation graph, considering the different influences of different relations and aggregating them as a comprehensive word feature. This process can be expressed as formula (4):

$$h_i^l = \sigma\left(\sum_{j\in V}\sum_{r\in R} P_r(i,j)\times\left(W_r^{l-1}h_j^{l-1} + b_r^{l-1}\right)\right) + h_i^{l-1} \quad (4)$$

where $P_r(i,j)$ represents the weight of the edge (words j to i are the probability of the relation r), W_r and b_r represent the GCN weight and bias vector of relation r, respectively. V and R represents all words and relation. More sufficient features are extracted for each word in this stage, making the effect of entity recognition and relation extraction becomes more obvious.

3.5 Combined output layer

This layer is a linear layer to integrate the information from the densely connected layer and eventually output the prediction results of GcnJereAT. The definition of the combined output layer is shown in formula (5):

$$h_{comb} = W_{comb}h_{out} + b_{comb} \quad (5)$$

where h_{out} represents the result obtained by integrating the outputs of independent dense layers, W_{comb} represents the weight matrix, b_{comb} represents the bias vector of the linear transformation.

3.6 Adversarial layer

In the field of NLP, there are two main types of adversarial samples generated by adding a perturbation η_{adv}: lexical level and character level. The lexical level mainly replaces words in sentences, such as synonyms, antonyms and adjectives. The character level is mainly achieved by modifying the words in the sentence, such as adding, deleting, and flipping. In addition, some constraints are added to maintain the basic meaning of syntax and sentence as much as possible. For example, the cosine similarity of two words is greater than 0.8, and the replacement of stop words is avoided. In this paper, the disturbance η_{adv} in the worst case is added to the original word embedding w by using the maximum loss function to generate confrontation samples, and the formula is as follows:

$$\eta_{adv} = argmax_{\|\eta\| \leq \varepsilon} \mathcal{L}(\omega + \eta; \hat{\theta}) \tag{6}$$

where $\hat{\theta}$ represents a copy of the model parameters. For the benefit of facilitating the use of formula (7) in neural networks, we use Goodfellow [12] method to approximate.

$$\eta_{adv} = \frac{\varepsilon g}{\| g \|} \tag{7}$$

$g = \nabla_{w} \mathcal{L}(\omega; \hat{\theta})$ is a differential of $\mathcal{L}(\omega, \hat{\theta})$, w is a differential operator, ε represents a small bounded norm considered as a hyperparameter, which is related to the dimension d of word embeddings. After generating the confrontation sample, the model is trained by mixing with the original sample, and the final maximum likelihood function is expressed by formula (8).

$$\mathcal{L}_{joint} = \mathcal{L}(\omega, \hat{\theta}) + \mathcal{L}(\omega + \eta_{adv}, \hat{\theta}) \tag{8}$$

4. Dataset and experiment

4.1 Dataset

We validate the performance of GcnJereAT using the NYT10 dataset [16], which is derived from 150 business reports, and it mainly includes 24 relations, a total of 61195 triples (56195 training sets, 5000 test sets). In addition, to validate the recognition effect of the model on relation overlapping and entity overlapping, the NYT10 dataset is divided into three types: normal entity (40729 triples), single-entity overlap (10760 triples) and entity-pair overlap (16032 triples).

4.2 Setting and Evaluation Criteria

Table 1 is the specific parameter settings. For the hyperparameter settings of the comparison models, we keep the same as the original paper.

Table 1

Specific settings of experimental parameters

Hyper-parameter	**Value**
Word embedding size	200
POS size	20
Bi-LSTM dropout rate	0.5
Bi-GCN layer	2
Epoch	100
Learning rate	0.001
Bias	10
Non-linear activation	ReLU
Bi-LSTM hidden size	200
Bi-LSTM layer	2
Optimizer	Adam

Evaluation criteria for entities: The result is regarded as correct only when the entity boundary and entity type are properly identified. Evaluation criteria for relationship types: The result can be considered correct only if both entities and their relationships are properly identified. We use precision, recall and F1 value as the evaluation criteria of entity and relation extraction results.

4.3 Comparative model

The following four types of joint extraction models are compared with our proposed GcnJere and GcnJereAT methods.

- **NTS** [6]: It is a sequence annotation-based model, which only uses sequence labeling not applicable to GCN and AT.
- **NGS** [9]: It is based on GCN rather than sequence labeling joint extraction and uses a loss function with bias weights to enhance the correlation between related entities.
- **MCM** [8]: It is an end-to-end model based on replication mechanism, which uses a dynamic encoder to extract relation.
- **CAT** [4]: It took full advantages of end-to-end and AT for joint extraction, which improved the robustness of the model.

4.4 Analysis

4.4.1 Overall comparative analysis of models

According to Table 2 it can be clearly conclude that the recall rate of GcnJere method in this paper is 25.5%, 15.1%, 17.2%, 0.6% higher than that of NTS, NGS, CAT, MCM, respectively, and F1 values increased by 17.8%, 8.9%, 9.5%, and 1.1%, respectively. More importantly, the recall rate of GcnJereAT method with the introduction of AT increased by 26.1%, 15.7%, 17.8% and 1.2%, respectively, and F1 values increased by 19.1%, 10.2%, 10.8% and 2.4%, respectively. Moreover, the precision of GcnJereAT method is at the top in the comparison models. The experimental results indicate that GcnJere and GcnJereAT are effective, and the detailed analysis is as follows.

Table 2

Experimental results on NYT10 dataset

Method	Precision	Recall	F1
NTS	62.6%	31.7%	42.0%
NGS	64.3%	42.1%	50.9%
CAT	69.2%	40.0%	50.3%
MCM	61.0%	56.6%	58.7%
GcnJere	65.7%	57.2%	59.8%
GcnJereAT	**71.1%**	**57.8%**	**61.1%**

The results showed in brackets are the average gains of GcnJere and GcnJereAT for the baselines. The bold represents the best method and the underlined represents the second-best method. The result denotes the improvement of GcnJere and GcnJereAT is significant based on the public benchmark NYT10.

1. **Comparison of joint extraction methods based on GCN and sequence labeling.**
   Compared to the NTS, the recall and F1 value of GcnJere are increased by 25.5% and 17.8%, respectively. Compared with CAT, the recall rate and F1 of GcnJereAT increased by 17.8% and 10.8%, respectively. In comparison with MCM, the accuracy rate, recall rate, and F1 value of GcnJere are increased by 0.7%, 0.6% and 1.1%. By comparing the experimental results, we can conclude that GCN has a better performance on the joint extraction task.
2. **Comparison of joint extraction methods based on GCN.**
   Both NGS and GcnJere methods extract entity relations based on graph convolutional networks. Compared with NGS, the recall rate and F1 value of GcnJere increased by 15.1% and 8.8%, respectively. In comparison with the NGS method, the primary reason for the improvement of GcnJere is that the pre-training of the text strengthens the quality of word embeddings and rebuilds a bi-directional graph structure based on the relation in the Densely Connection layer, which captures the local information and non-local information more fully. The NGS method only uses a directed graph to train itself, so the performance of NGS has a certain gap with GcnJere.
3. **Comparison of joint extraction methods based on GCN and AT.**
   NGS, GcnJere, and GcnJereAT all adopt the GCN while GcnJereAT adds AT on this basis to train the model with a slight disturbance. Compared with the first two models, precision, recall rate, and F1 value of the GcnJereAT model increased by 1.9%~5.4%, 0.4%~15.7%, and 1.3%~10.2%, respectively. Outcome of the experiment indicate that AT could increase the robustness of the model, which verifies the effectiveness of AT in joint extraction tasks.

4.4.2 Function analysis of each part of the model

In addition, we remove different parts of GcnJereAT to verify the impact of each part on its performance.

As shown in Table 3, we can conclude that the four parts of pre-training, Bi-LSTM, pruning dependency tree, and AT have an important impact on the joint extraction model of the GCN proposed in this paper.

Table 3

Effect of different components on model performance

Method	**PT**	**LSTM**	**DT**	**AT**	**F1**
Best	√	√	√	√	61.1%
-PT	×	√	√	√	58.9%
-LSTM	√	×	√	√	57.2%
-DT	√	√	×	√	58.3%
-AT	√	√	√	×	59.8%

Note: '×' indicates that this step is not used, PT indicates pre-training, DT indicates dependency tree without running, and AT indicates adversarial training.

- Without pre-training, the F1 value of the model decreased by 2.2%. Using SynGCN pre-training, we improved the quality of word embeddings and had a less negative impact on subsequent data processing steps. Therefore, high-quality data is an important part of the training model.
- Without the Bi-LSTM layer, the F1 value of the model decreased by 3.9%. The introduction of Bi-LSTM has the characteristics of long-distance coding, which can make up for the deficiency of GCN in long distance relation extraction.
- Without pruning on the dependency tree, the F1 value decreased by 2.8%. Unpruned dependency trees produce redundant information, which makes the experimental results decline and affects the extraction effect of the model. In addition, since redundant information is not needed to be processed after pruning, the calculation speed of the model is improved as well.
- Without AT, the F1 value decreased by 1.3%. In the original input data, some disturbances that affect the model effect. AT and the addition of confrontation samples enhances the distinguish rate of the model to the input disturbance. Therefore, after the removal of AT, the F1 value is also reduced.

4.4.3 Analysis of entity and relation overlap

To validate the extraction effect of the GcnJereAT model in entity overlapping and relation overlapping, we compare the specific two models of MCM in this paper. Among them, one represents the basic model of MCM, which only uses a single decoding layer to extract entity and its relation, and Multi represents the improved model of MCM which extracts entity and its relation dynamically by using multiple decoding layers. The comparison results are shown in Figure 2.

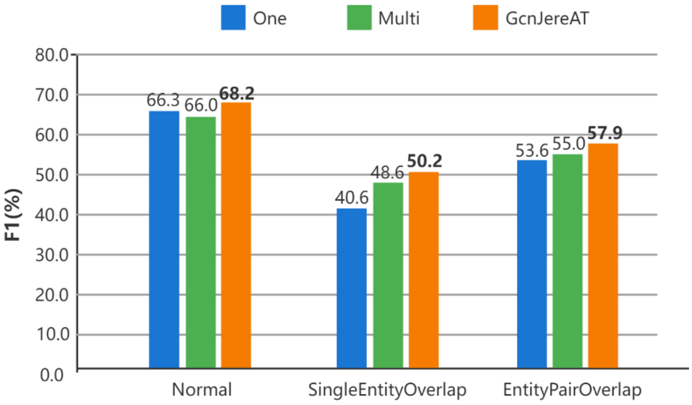

Fig. 2. Comparison of F1 values of the models

The F1 of the GcnJereAT model is 1.6%~9.6% higher than that of One and Multi models in normal relation, entity overlapping, and relation overlapping. The F1 value of the GcnJereAT model improves 1.6%~2.9% compared to the Multi model, even though the Multi model uses dynamic decoding to improve the performance of the One model. The overall performance of GcnJere is superior to the MCM model, which verifies the effectiveness of GcnJere in settling entity overlap and relation overlap.

5. Conclusions

In this paper, we proposed two joint entity-relationship extraction models. The first model is called GcnJere, which simply introduces GCN. The other model, called GcnJereAT, is the research focus model in this paper, which introduces GCN and adversarial training to effectively address the information redundancy, propagation errors and information loss problems in traditional pipeline methods. Our models also improve the handling of entity overlap and relationship overlap, as well as the robustness of the models. The follow-up plan aims to improve the proposed model, for example, by adopting a new and better adversarial training approach to further improve the performance of the GCN-based model.

Acknowledgements

This work was supported by the CACMS Innovation Fund (CI2021A05403&CI2021A00512) and Beijing Forestry University Student Innovation and Entrepreneurship Training Plan (X202110022185).

REFERENCES

[1]. *Sun C Z, Gong Y Y, Wu Y B, et al.* Joint type inference on entities and relations via graph convolutional networks. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 1361-1370.

[2]. *Hong Y, Liu Y, Yang S, et al.* Joint extraction of entities and relations using graph convolution over pruned dependency trees. Neurocomputing, 2020, 411: 302-312.

[3]. *Bekoulis G, Deleu J, Demeester T, et al.* Adversarial training for multi-context joint entity and relation extraction. arXiv preprint arXiv:1808.06876, 2018.

[4]. *Huang P, Zhao X, Fang Y, et al.* End-to-end Knowledge Triplet Extraction Combined with Adversarial Training. Journal of Computer Research and Development, 2019, 56(12): 2536-2548.

[5]. *Li Q, Ji H.* Incremental Joint Extraction of Entity Mentions and Relations. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014: 402-412.

[6]. *Zheng S C, Wang F, Bao H, et al.* Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075, 2017.

[7]. *Bekoulis G, Deleu J, Demeester T, et al.* Joint entity recognition and relation extraction as a multi-head selection problem. Expert Systems with Applications, 2018, 114: 34-45.

[8]. *Zeng X, Zeng D, He S, et al.* Extracting relational facts by an end-to-end neural model with copy mechanism. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 506-514.

[9]. *Wang S, Zhang Y, Che W, et al.* Joint extraction of entities and relations based on a novel graph scheme. IJCAI. 2018: 4461-4467.

[10]. *Yan Z, Zhang C, Fu J, et al.* A partition filter network for joint entity and relation extraction. arXiv preprint arXiv:2108.12202, 2021.

[11]. *Szegedy C, Zaremba W, Sutskever I, et al.* Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[12]. *Goodfellow I J, Shlens J, Szegedy C.* Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[13]. *Miyato T, Dai A M, Goodfellow I.* Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725, 2016.

[14]. *Yasunaga M, Kasai J, Radev D.* Robust multilingual part-of-speech tagging via adversarial training. arXiv preprint arXiv:1711.04903, 2017.

[15]. *Vashishth S, Bhandari M, Yadav P, et al.* Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. arXiv preprint arXiv:1809.04283, 2018.

[16]. *Riedel, Sebastian, Limin Yao, and Andrew McCallum.* Modeling relations and their mentions without labeled text. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2010.