6-2023

Multi-head attention graph convolutional network model: End-to-end entity and relation joint extraction based on multi-head attention graph convolutional network

Zhihua TAO

Chunping OUYANG

Yongbin LIU

Tonglee CHUNG

Yixin CAO
Singapore Management University, yxcao@smu.edu.sg

Citation

TAO, Zhihua; OUYANG, Chunping; LIU, Yongbin; CHUNG, Tonglee; and CAO, Yixin. Multi-head attention graph convolutional network model: End-to-end entity and relation joint extraction based on multi-head attention graph convolutional network. (2023). *CAAI Transactions on Intelligence Technology*. 8, (2), 468-477.
Available at: https://ink.library.smu.edu.sg/sis_research/9347

Received: 20 November 2020 | Revised: 13 February 2022 | Accepted: 8 March 2022

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12086

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Multi-head attention graph convolutional network model: End-to-end entity and relation joint extraction based on multi-head attention graph convolutional network

Zhihua Tao[1] | Chunping Ouyang[1] | Yongbin Liu[1] | Tonglee Chung[2] | Yixin Cao[3]

[1]School of Computer Science, University of South China, Hengyang, China

[2]Department of Computer Science and technology, Tsinghua University, Beijing, China

[3]School of Computer and Information Systems, Singapore Management University, Singapore, Singapore

Correspondence

Chunping Ouyang, University of South China, No. 28, Changsheng West Road, Hengyang City, Hengyang, 421001 Hunan, China.
Email: ouyangcp@126.com

Funding information

State Key Program of National Natural Science of China, Grant/Award Number: 61533018; National Natural Science Foundation of China, Grant/Award Number: 61402220; Philosophy and Social Science Foundation of Hunan Province, Grant/Award Number: 16YBA323; Natural Science Foundation of Hunan Province, Grant/Award Number: 2020JJ4525; Scientific Research Fund of Hunan Provincial Education Department, Grant/Award Numbers: 18B279, 19A439

Abstract

At present, the entity and relation joint extraction task has attracted more and more scholars' attention in the field of natural language processing (NLP). However, most of their methods rely on NLP tools to construct dependency trees to obtain sentence structure information. The adjacency matrix constructed by the dependency tree can convey syntactic information. Dependency trees obtained through NLP tools are too dependent on the tools and may not be very accurate in contextual semantic description. At the same time, a large amount of irrelevant information will cause redundancy. This paper presents a novel end-to-end entity and relation joint extraction based on the multi-head attention graph convolutional network model (MAGCN), which does not rely on external tools. MAGCN generates an adjacency matrix through a multi-head attention mechanism to form an attention graph convolutional network model, uses head selection to identify multiple relations, and effectively improve the prediction result of overlapping relations. The authors extensively experiment and prove the method's effectiveness on three public datasets: NYT, WebNLG, and CoNLL04. The results show that the authors' method outperforms the state-of-the-art research results for the task of entities and relation extraction.

KEYWORDS

information retrieval, natural language processing

1 | INTRODUCTION

Entities and relations extraction are two important subtasks of information extraction. The purpose is to extract the semantic relations between entity pairs from unstructured text. Usually, the relation between entity pair is visually described as a triplet (e.g. China, capital, Beijing). Currently, the entity and relation extraction methods are mainly divided into pipeline learning and joint learning.

The pipeline methods are used to extract the relations on the basis of entity recognition proposed by Kambhatla [1] and Fundel [2]. However, in these methods, relation extraction completely depends on the accuracy of entity recognition, which may lead to error propagation. The emergence of joint learning methods based on deep learning synchronises entity recognition and relation extraction proposed by Miwa [3], Li [4], and Giannis [5], which makes full use of the interactivity.

Miwa and Bansal [6] propose a Bi-directional Long-Short Term Memory (Bi-LSTM) to automatically learn features and use a dependency tree to model and extract relations. LSTM is a kind of recurrent neural network (RNN). Its main function is to carry out long-term memory for data information. It cannot encode from back to front, while Bi-LSTM can do it. For the NER task, some entities require a model that can capture long sequence features; BiLSTM-conditional random field (CRF) is the mainstream model [7]. However, due to the disadvantage of gradient diffusion, Wei et al. [8] later introduced attention mechanisms to increase memory ability. Gupta et al. [9] show a method of extracting features using NLP tools and RNN. Later, a graph convolutional network (GCN) model based on pruning dependent trees was proposed by Zhang [10]. Guo et al. [11] attempt relation extraction tasks by combining GCN and dependency trees. However, these methods rely too much on NLP tools, which is used to construct dependency trees or tag part-of-speech (POS). NLP tools have a strong dependence on the grammar of the text. The better the grammatical structure of the text, the higher the analysis performance. In contrast, irregular text structure leads to reduced results. The purpose of joint extraction is to avoid that the latter task relies too much on the result of the former task. Although these methods avoid the drawbacks of the pipeline methods, the results of the methods rely on the accuracy of the NLP tools parse, not a real end-to-end joint model. Figure 1 shows the wrong dependency analysis of the NLP tool. MAGCN uses spaCy [12] analysis in this sentence. There are two 'ROOT' in a sentence. This violates the first constraint axiom of the dependency parsing tree: only one word (root) that does not depend on other words.

Overlapping relations are significant problems in entity and relation extraction. Overlapping relations mean relations are sharing common entities or entity mentions. The relation overlap types are divided into normal, entity pair overlap (EPO), and single entity overlap (SEO). Table 1 shows the different relation overlap types. Compared to normal relations, EPO and SEO overlap types are harder to extrapolate and obtain. Although a new tagger strategy method proposed by Zheng [13] merges entity and relation extraction into sequence tagger, which solves the problem of entity redundancy caused by parameter sharing; it still has not solved the overlapping problem. Bekoulis et al. [14] use LSTM in the joint model without parameter sharing. However, the task of relation extraction is treated as a single-head selection problem, and it means dealing with one relation at a time. There is a direct or indirect relation between entities. How to transfer interactive information is still challenging to solve.

For the problems in the joint extraction model, we propose an end-to-end joint extraction model based on multi-head attention graph convolution (MAGCN). MAGCN obtains the adjacency matrix through the multi-head attention and combines it with graph convolution networks. This method does not require any NLP tools to extract features or dependency trees. While the joint extraction is completed, multiple relations are processed simultaneously. In the MAGCN model, our main contributions are as follows:

- Our end-to-end model does not need external tools to obtain the features and the dependency tree. It obtains the features through the multi-head attention combined with GCN, which avoids errors propagation.
- Our model uses multi-head selection to make entities match multiple relations. Instead of finding the head word for each word and then matching a possible relation between them, we synchronously pair the possible head word and relation for each word. It means that each word may have different head words and relations.
- The experimental results show that our method outperforms other methods. In addition, we prove that the adjacency matrix extracted by the model has better recognition accuracy than external tools.

This paper is organised as follows: we describe the related work in Section 2. Section 3 presents the proposed model and experiments in Section 4. We conclude this paper in Section 5.

2 | RELATED WORK

For early pipeline methods, which implemented entity recognition and relations extraction separately, that is, after named entity recognition [15] is completed, the relation extraction was carried out by Bach [16]. Still, the pipeline methods have some

TABLE 1 Relation overlap types

Relation type	Example	True result
Normal	In Brussels, European Union leaders had approved a new budget	(European Union, contains, Brussels)
EPO	But at least 20 cities in Norway have longer nights than Oslo	(Norway, capital, Oslo) (Norway, contains, Oslo)
SEO	His mother edits book reviews for Wellesley magazine, the alumnae publication for Wellesley college in Massachusetts.	(Massachusetts, contains, Wellesley) (Wellesley, contains, Wellesley college)

Abbreviations: EPO, entity pair overlap; SPO, single entity overlap.

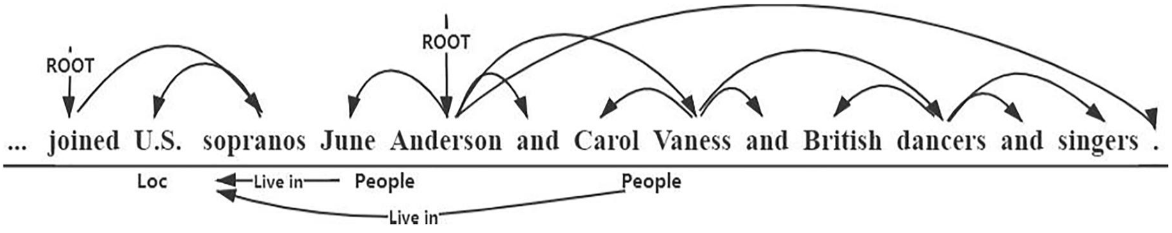
ROOT
ROOT
... joined U.S. sopranos June Anderson and Carol Vaness and British dancers and singers .
Loc
Live in
People
People
Live in

FIGURE 1 Wrong parsing result

drawbacks: first, the errors of entity recognition will directly influence the accuracy of relation extraction; second, the relation between two subtasks is ignored, resulting in information loss; third, redundant information is brought by unrelated entities. For these reasons, joint extraction methods have been developed.

For joint entity and relation extraction, Miwa and Sasaki [3] propose a manually extracted feature method for named entity recognition and relation extraction tasks. This method requires complex feature engineering and relies heavily on NLP tools, which may lead to error propagation problems. With the popularity of GCN, the researchers began to consider the dependency structure. Fu et al. [17] apply the GCN to an end-to-end relation extraction model. They use the NLP tool (spaCy) to get POS tag and dependency tree and combine RNN and GCN to extract sequential and regional features. Marcheggiani and Titov [18] propose GCN semantic role annotation combining grammatical information in the sequence model. But these methods still rely on NLP tools to get POS tag or dependency tree construct adjacency matrix. It's difficult for the tools to parse complex or non-standard text, and incorrect parsing directly affects the accuracy of extraction results.

Zeng et al. [19] propose an end-to-end model based on sequence-to-sequence learning. They use a unified decoder or several independent decoders in the process of decoding relational triples. The decoder forms triples by selecting a relation and coping entities in the text and captures the interaction of the relations without direction. However, this does not completely solve the problem of overlapping. When multiple entity pairs have the same relation, they cannot be solved. Fei et al. [20] treat the task as a quintuple prediction problem to construct the overlapping relation extraction model. Bai et al. [21] also use end-to-end model based on a double pointer networks to improve the performance of relation extraction. Katiyar and Cardie [22] propose a Bi-directional Long Short-Term Memory (Bi-LSTM)-based model for the joint task and encode the whole sentence, but identify the tokens in a relation with the other tokens. Zheng et al. [13] convert the joint extraction task into a tagging problem. Also, Bi-LSTM is used to encode the input sentences and the decoding layer with bias loss, which can enhance the correlation of entity tags. However, this method only considers the case, where an entity belongs to a triplet, and does not consider the effect of overlapping relations.

In our proposed model, the MAGCN constructs an adjacency matrix through attention mechanisms without relying on NLP tools to obtain feature information. It is a complete end-to-end joint extraction model. And it uses head relation extraction to take all word pairs into account.

3 | MODEL

This section provides an overview of the complete modelling method, with detailed descriptions of implementing the attention and GCN layers.

Algorithm 1 Complete algorithm with MAGCN

```
Input: sentence $L$: $(l_1, l_2, ..., l_n)$; $K$:
    relations num; $T$: Layers of Graph
    convolution; $L_{ner}$: Loss of ner; $L_{rel}$: Loss
    of relation; $w_1, w_2, w_3$: initialisation
    parameter;
Output: distribution $P$
1: initial $x_v, v \in n \leftarrow W$;
2: $H_v^0 \leftarrow x_v,\ v \in n$
3: for $i = 1; i \leq n$ do
4:      head $\leftarrow$ get Attention$(W_1 H, W_2 H, W_3 H)$
5:      adjancy matrix $A = $ concat$(\text{head}_1, ...,$
        $\text{head}_8) W_k$
6:      for $t = 1; t \leq T$ do
7:          for $k = 1; k \leq K$ do
8:              $h_{ik}^{t+1} \leftarrow$ get Graph$\left(A_{ij}^k, h_{ik}^t, W, b\right)$
                $(j \in n)$
9:              $H^{t+1} \leftarrow$ get node information
                $\left(h_k^{t+1}, H^t\right)$
10:         end for
11:     end for
12:     calculate the score of entities
        Score $(e_i)$
13      calculate the score between the
        entities under the relation $k$ :
        Score$\left(e_i, k, e_j\right)$
14: get predict $P(e_i), P\left(e_i, k, e_j\right)$
15: end for
16: $L(\theta) = L_{ner} + L_{rel}$
17: update by minimising the loss
18: return
```

The complete structure diagram of our model is shown in Algorithm 1 and Figure 2. In Algorithm 1, MAGCN obtained the feature representation of each word in the input sentence (lines 4–9), calculated the entity score and the relation score between entity pairs (lines 12, 13), respectively, and finally obtained the prediction result. The model implementation process is shown in Figure 2. First, MAGCN combines char embedding and word embedding as initial input and uses Bi-LSTM to extract context word features and then uses attention mechanism to construct R weighted adjacency matrices, where R is the number of relations between entities. After, extract the regional features of words under each relational adjacency matrix by GCN. Finally, MAGCN uses a CRF to complete the prediction of entities. The model adopts a head selection in joint extraction. Each entity may have multiple relations with other entities, so we obtain the scores between entities under each relation.

3.1 | Encode layer

In the encoding layer, the model first maps sentence $l_1, ..., l_n$ as token sequence to word vector. Here we use the combination of the pre-training word embedding and character embedding as the initial embedding. Hochreiter proposed Bi-

LSTM [23], a model that can capture long-distance context information. It encodes the sequence from left to right and right to left to obtain the bidirectional information representation of each word. We obtain the context representation t_i by passing the vector x_i into Bi-LSTM. The initialisation vector is composed by

$$x_i = \text{char} \oplus \text{word} \tag{1}$$

The hidden feature representation of the ith word is expressed as

$$\overrightarrow{m_i} = \overrightarrow{\text{LSTM}}\left(\overrightarrow{m_{i-1}}, x_i\right) \tag{2}$$

$$\overleftarrow{m_i} = \overleftarrow{\text{LSTM}}\left(\overleftarrow{m_{i-1}}, x_i\right) \tag{3}$$

$$m_i = \left[\overrightarrow{m_i}; \overleftarrow{m_i}\right] \tag{4}$$

where $\overrightarrow{m_i}$ and $\overleftarrow{m_i}$ are the hidden state in the forward and backward LSTM.

3.2 | Multi-head attention layer

In the multi-head attention layer, we transform the word features into the relation adjacency matrix A. Figure 3 shows the process of operation at attention layer. We first expand the hidden feature $M \in \mathbb{R}^{n\times d}$ obtained in the encode layer into a multi-dimensional vector $M^{\sim} \in \mathbb{R}^{n\times n\times d}$, n is the number of words in a sentence and d is the hidden size of Bi-LSTM. Multi-head attention can handle information from different

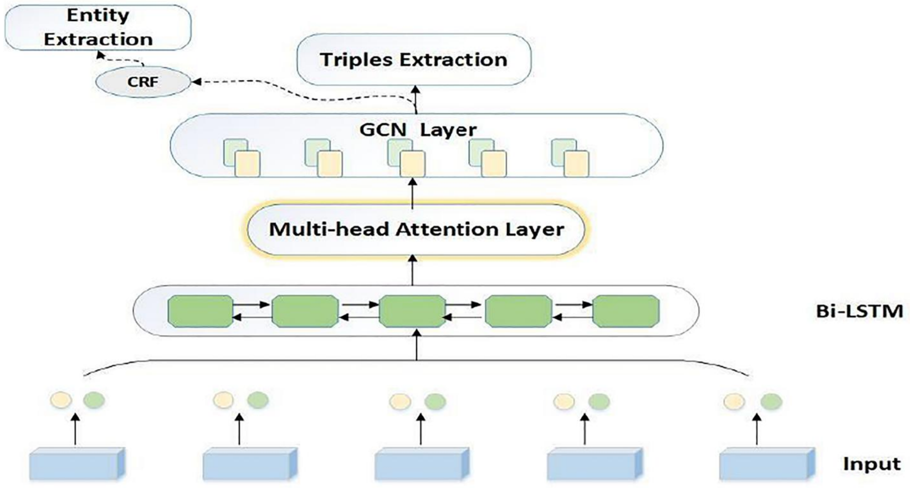

FIGURE 2 Model for joint entity and relation extraction with multi-head attention graph convolutional networks

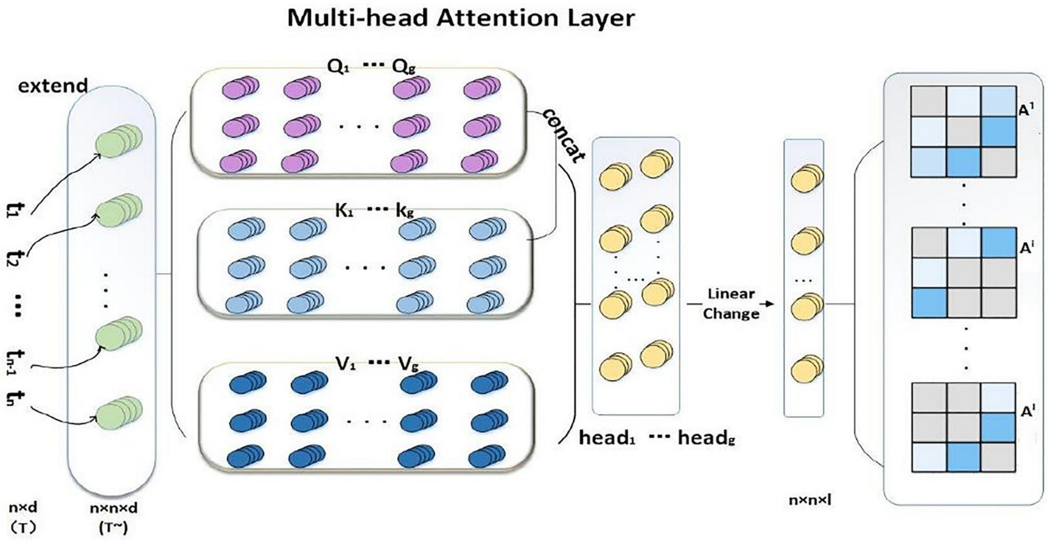

FIGURE 3 The architecture of multi-head attention layer

subspaces. Attention calculation requires query and key from the same target, where E is the input hidden representation,

$$\mathrm{E} = M^{\sim} W_e + b_e \tag{5}$$

Then, we get the ith head as follows, we perform 'concat' operation on both Q and K. Softmax function is applied to get the weight values of attention,

$$Q_i = \mathrm{E}W_i^{Q} \tag{6}$$

$$K_i = \mathrm{E}W_i^{k} \tag{7}$$

$$V_i = \mathrm{E}W_i^{v} \tag{8}$$

$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{W_i[Q_i : K_i]}{\sqrt{d}}\right) V_i \tag{9}$$

where $i \in \mathbb{R}^g$, g is the number of attention head. $W_e \in \mathbb{R}^{d\times d}$, $b_e \in \mathbb{R}^d$, W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{d\times c}$ are learnt weights and $c = d/g$. The g weight heads are concatenated together, and then linear transformation is performed with the new weight matrix W. The output is shown in Equation (10), where $W \in \mathbb{R}^{gc\times d}$,

$$A^{\sim} = \left[\mathrm{head}_i; \mathrm{head}_2; \ldots; \mathrm{head}_g\right] W \tag{10}$$

where $W \in \mathbb{R}^{gc\times l}$, and l denotes the relation dimension. $A^{\sim} \in \mathbb{R}^{n\times n\times l}$ is the adjacency matrix for GCN input. Concretely, $A^{\sim}$ contains L sub-matrices, A^m represents the fully connected graph under the relation m, and A_{ij}^{m} represents the weight of the nodes i, j (i, j under the relation m).

3.3 | GCN layer

GCN [24] is an extension of CNN coding graph structure, which is used to obtain the structural information of adjacency nodes on the graph. The original Bi-GCNs consider the bidirectional node information, and the adjacency matrix A represents the information between nodes, where there is an edge between nodes i to j and then $A_{ij} = 1$. Figure 4 shows the specific operation in the GCN layer. A^i is expanded from the original adjacency matrix to the relational weighted matrix after the attention layer. The darker the colour in the matrix, the greater the weight of attention between the two nodes. A_{ij} and A_{ji} have different meanings. The former means that nodes i to j have an edge, and the latter mean nodes j to i have an edge. Under each relation, we have an adjacency matrix and its form is the same as the original adjacency matrix as $n \times n$. We aggregate the information of words based on each relation. We get the outgoing edges and the incoming edges from each node under each relation and concentrate them (initialisation feature h_0 is the output of the encoding layer),

$$\overleftarrow{h_{i*}^{t+1}} = ReLU\left(\sum_{j=1}^{n} A_{ij*} \overleftarrow{W^t} h_{j*}^t + \overleftarrow{b^t}\right) \tag{11}$$

$$\overrightarrow{h_{i*}^{t+1}} = ReLU\left(\sum_{j=1}^{n} A_{ij*} \overrightarrow{W^t} h_{j*}^t + \overrightarrow{b^t}\right) \tag{12}$$

$$h_{i*}^{t+1} = \left[\overleftarrow{h_{i*}^{t+1}}, \overrightarrow{h_{i*}^{t+1}}\right] \tag{13}$$

where $\overleftarrow{W}, \overrightarrow{W}$ and $\overleftarrow{b}, \overrightarrow{b}$ mean outgoing and incoming weights separately. A_{ijk} and $A_{ijk} \in \mathbb{R}^{n\times n}$ mean existence path from i to j and the path from j to i under the relation k. $\overleftarrow{h_{i*}^{t+1}}$ and $\overrightarrow{h_{i*}^{t+1}}$ represent the outgoing and incoming features, respectively. Then, concatenate both outgoing and incoming features as the final feature. h_{ik}^{t+1} represents the hidden features of the word i under the relation k at layer $t+1$, $\mathrm{i} \in \mathbb{R}^n$, $k \in \mathbb{R}^l$. Therefore, we obtain the feature representation results $h_{i1}, h_{i2}, \ldots, h_{il}$ under the l relations layer. H is the feature

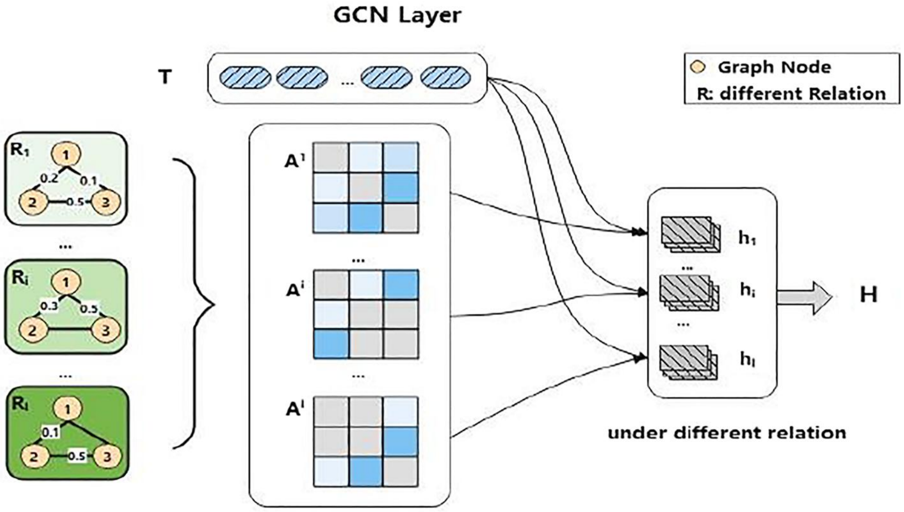

FIGURE 4 The architecture of graph convolutional network layer

consists by each node under each relation. Use the previous layer's representation features and the results of this layer to update the word representation as a new round of input.

$$H^{t+1} = \sum_{j\in l} h_{*j}^{t+1} + H^{t} \tag{14}$$

$h_{*j} \in \mathbb{R}^{n\times d}$ represents all the node information under relation j; H^{t+1} represents the final output at layer $t+1$.

3.4 | Prediction

In the entity prediction layer, we use the BIO (Begin, Inside, Outside) encoding scheme and CRF to calculate each word's probable tag. And for the joint extraction, since each entity may have different relations with other entities, we present the joint extraction as a multi-head selection proposed by Zhang [25] and Bekoulis [14]. For each word, we predict the head and relation simultaneously. When each word is given, the weight score under the relation r with other words (treated as head) will be predicted, and the probability of the words will be calculated,

$$P_r(e_i, r_k, e_j) = \mathrm{softmax}\big(f\big(\mathrm{T}H_i + \mathrm{W}H_j + b\big)\mathrm{U}\big) \tag{15}$$

where e_i, e_j represents entities i and j; r_k is the relation k. Where $f(.)$ is an activation function (tanh), and T, W, U are all weights.

Last, we use the cross-entropy loss for both entity and relation during training. For the given word, we use BIO tagging to choose the highest score, that is, each word judged belongs to one of these categories. And the total loss is given as loss= $\mathrm{loss}_{\mathrm{ner}}$+$\mathrm{loss}_{\mathrm{rel}}$.

4 | EXPERIMENTS

In this section, we will present the experimental results of our model. We first explain the datasets and settings used in the experiments. Next, the baselines used for the comparative experiments will be described. Finally, the results on the datasets will be explained. In addition, to prove the model's effectiveness, we add additional experiments.

4.1 | Datasets and settings

We evaluate the performance of our model on three datasets: NYT [26], WebNLG [27], and CoNLL04 [28]. We use NYT and WebNLG to do the baseline comparative experiments and use CoNLL04 to do the additional experiment of whether NLP tools are used to construct dependency trees for better results. As baselines, we keep sentences with less than 100 words for NYT and the first sentence for WebNLG. First, we set different parameters and select the best value to apply to our model. Learning rate is an important index to improve the accuracy of the model. We set the learning rate of different parameters for comparison. As shown in Figure 5, when the learning rate is 0.001 on NYT and WebNLG datasets, the MAGCN model has the best effect. In the same way, we compare the parameters under different dimensions of Bi-LSTM and GCN. Since the input and output dimensions do not match, information will be lost, the dimensions of Bi-LSTM and GCN will be changed synchronously.

In our experiment settings, we used the pre-trained GloVe as word embedding (dimension 300). The initial input of words was concatenated to word embedding and char embedding (dimension 25). The dimension for the Bi-LSTM is 256 and for the Bi-GCN is 256. We set the head number of attention to 8. The optimiser Adam [29] and the learning rate is 0.001. The specific settings are shown in the Table 2.

4.2 | Baselines

We compare our method with the following baseline methods to improve the performance: Zheng [13] proposed Novel-Tagging, which predicts entity and relation through sequence tagger. CopyRe-Multi proposed by Zeng [19] adopts the dynamic decoders and construct an end-to-end neural model and use multi-decoder. CopyRe-One uses a single decoder to

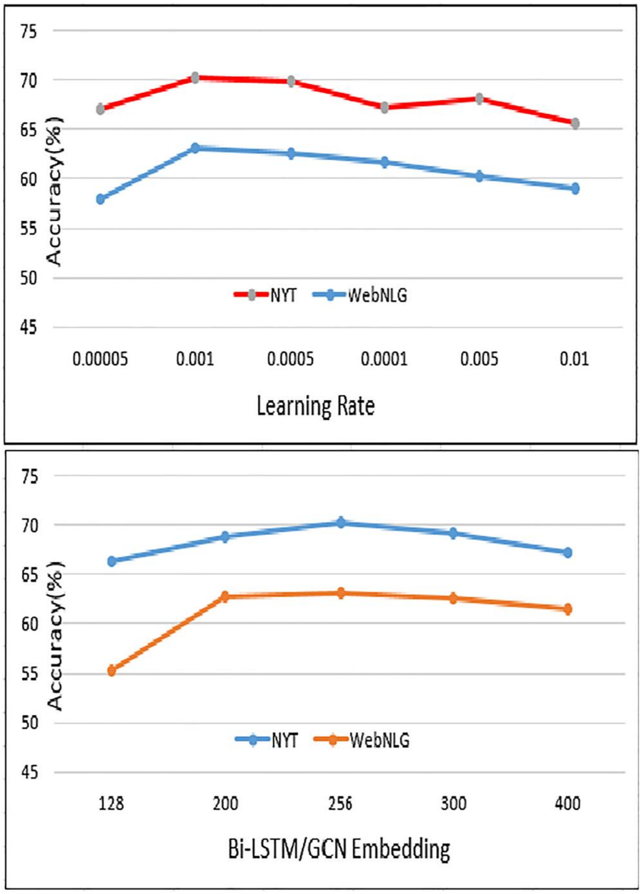

FIGURE 5 Accuracy changes with the increase of learning rate and Bi-LSTM/GCN embedding dimension on the two datasets

extract relation, which is proposed in Multi-Decoder. GraphRel proposed by Fu [17] uses graph convolutional networks to joint learn entity and relation phase by phase and uses linear and dependency structures to extract sequential and regional features. Fei [20] proposed BER, which employed a graph attention model to enhance the interactions between overlapping triplets. SPointer used multiple decoders to predict the entity with the start and end position, which is proposed by Bai [21].

5 | Results and analysis

Table 3 shows comparative results for our method and all baselines. We present the result in precision(p), recall(R) and F1. The calculation formula is as follows:

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{16}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{17}$$

$$\mathrm{F1} = \frac{2 * P * R}{P + R} \tag{18}$$

TP (True Positive) represents the number of positive samples predicted to be positive samples, TN (True Negative) represents the number of negative samples predicted to be negative samples, and FP (False Positive) represents the number of positive samples predicted to be negative samples.

TABLE 2 The parameters set of MAGCN

Word embedding dimension	300
Char embedding dimension	25
Bi-LSTM dimension	256
Bi-GCN dimension	256
Optimiser	Adam
Learning rate	0.001

Abbreviation: MAGCN, multi-head attention graph convolutional network model

TABLE 3 Results for our method and all baselines on datasets

	NYT			WebNLG		
Method	**p**	**R**	**F1**	**p**	**R**	**F1**
NovelTagging	62.4%	31.7%	42.0%	52.5%	19.3%	28.3%
CopyRe-one	59.4%	53.1%	56.0%	32.2%	28.9%	30.5%
CopyRe-multi	61.0%	56.6%	58.7%	37.7%	36.4%	37.1%
GraphRel	63.9%	60.0%	61.9%	44.7%	41.1%	42.9%
BER	57.2%	56.9%	56.5%	-	-	-
SPointer	72.8%	**69.0%**	**70.9%**	38.7%	37.5%	38.1%
MAGCN(ours)	**85.2**%	59.7%	**70.2**%	**84.6**%	**50.3**%	**63.1**%

Note: These bolds values mean that the comparison result is the best.

Our model has a large jump in performance: For the NYT dataset, we observe that our model outperforms NovelTagging by 28.2%, CopyRe-One by 14.2%, CopyRe-Multi by 11.5%, GraphRel by 8.3%, and for BER, we outperform it by 13.7% for F1. We have also gotten great results on the WebNLG. We outperform NovelTagging by 34.8%, CopyRe-One by 32.6%, CopyRe-Multi by 26%, GraphRel by 20.2% and SPointer by 25%.

NovelTagging uses LSTM decoder and realises entity pairs only belong to one relation, ignoring the overlap relations. So it has high precision and low recall. CopeRe-One and CopyRe-Multi adopt the seq2seq mechanism and predict one triplet for each decoder, so they can only extract a finite number of triples. GraphRel solves the problem of the entity overlap while building triplets and combining them with an external dependency tree. Relying on external tools to create dependencies can affect the correct semantic relations to some extent. So it will not work well for the WebNLG dataset with a wide variety of relations. BER uses a multi-layer convolution and self-attention mechanism as the encoder and uses the double-pointer to identify the whole entity, ignoring overlapping relations. Although SPointer's results on NYT are higher than that of ours, it is greatly influenced by data; it has a slight disadvantage in complex relations and insufficient stability. Our results show that our model has good stability. We believe the result is that constructing graph convolutions through the characteristics of the sentence itself can obtain more accurate information, which shows that external tools have some disadvantages.

Here, we perform further analysis on the model we proposed. We present the results of different triplet types, the entity recognition result, and the effect of the adjacency matrix constructed by internal and external NLP tools.

As per GraphRel, relation triplets are divided into Normal, EPO and SEO. The statistics are shown in Table 4. The F1 results under different relation triplets are shown in Figure 6. In normal relation, MAGCN outperforms CopyRe-One by 15.7%, CopyRe-Multi by 16% and GraphRel by 12.4% on NYT; it outperforms CopyRe-One by 11.4%, CopyRe-Multi by 19.2% and GraphRel by 12.6% on WebNLG. In EPO, MAGCN outperforms GraphRel by 5% on NYT and by 19.7% on WebNLG, outperforms CopyRe-One and CopyRe-Multi by 9.6% on NYT and by 8.2% on WebNLG. We suppose CopyRe-One and CopyRe-Multi predict only one triplet,

TABLE 4 Statistics of datasets

	NYT		WebNLG	
Category	**Train**	**Test**	**Train**	**Test**
Relation	24		246	
Normal	37013	3266	1596	246
EPO	9782	978	227	26
SEO	14735	1297	3406	457

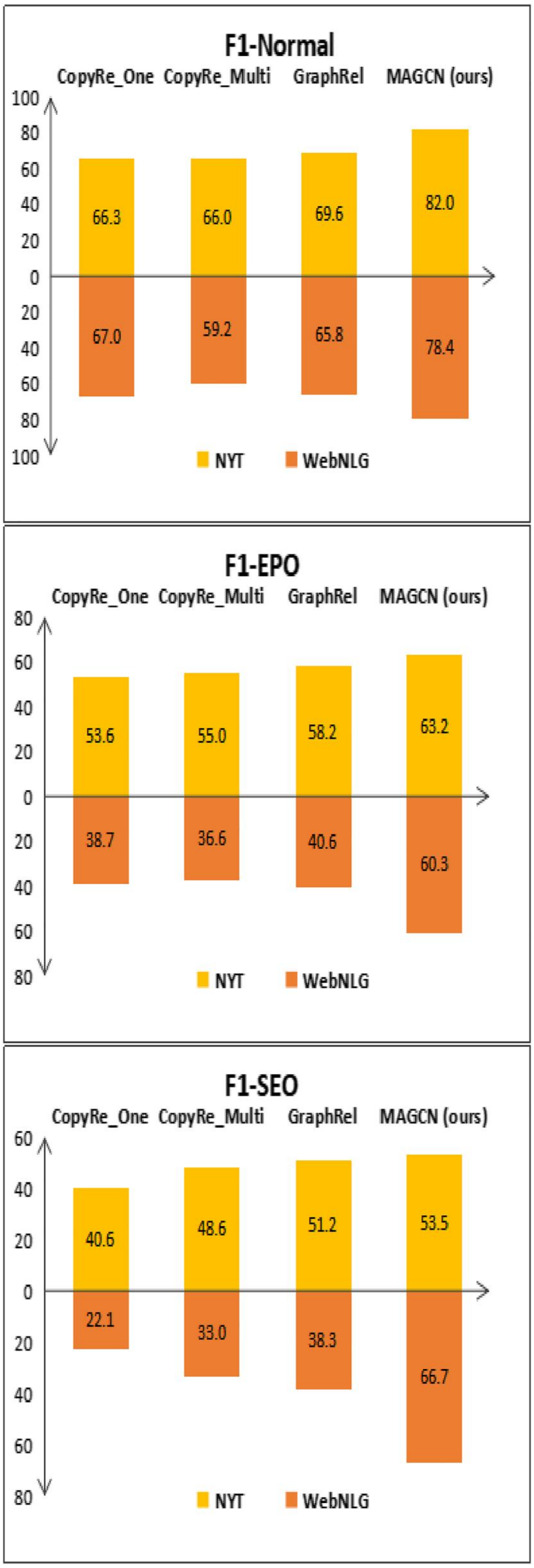

FIGURE 6 Results of different relation types

which is not good for overlapping relations. And in SEO, it outperforms CopyRe-One by 12.9% on NYT and by 44.6% on WebNLG, outperforms CopeRe-Multi by 4.9% on NYT and by 33.7% on WebNLG, outperforms GraphRel by 2.3% on NYT and by 27.8% on WebNLG. Through this experiment, it can be proved that our method in the problem of overlapping relations is significantly improved compared with that of the previous method.

TABLE 5 F1 results of entity recognition and relation extraction separately

NYT				
		Relation		
Method	ner	Normal	EPO	SEO
CopyRe-one	64.7%	66.3%	53.6%	40.6%
CopyRe-multi	75.6%	66.0%	55.0%	48.6%
GraphRel	89.2%	69.6%	58.2%	51.2%
MAGCN (ours)	**94.2%**	**82.0%**	**63.2%**	**53.5%**
WebNLG				
		Relation		
Method	ner	Normal	EPO	SEO
CopyRe-one	59.5%	67.0%	38.7%	22.1%
CopyRe-multi	78.2%	59.2%	36.6%	33.0%
GraphRel	91.9%	65.8%	40.6%	38.3%
MAGCN (ours)	**94.7%**	**78.4%**	**60.3%**	**66.7%**

Note: These bolds values mean that the comparison result is the best.

Abbreviations: EPO, Entity Pair Overlap; MAGCN, multi-head attention graph convolutional network model; SEO, Single Entity Overlap.

In addition, we do comparative experiments for the entity recognition task. We show the results of the entity task and the relational task experiment together in Table 5. We get a 29.5% and 18.6% improvement on NYT compared to CopyRe-one and CopyRe-Multi, outperforming GraphRel by 5% for entity recognition. For WebNLG, we obtain better result than the best result (GraphRel) with 2.8% for entity recognition. Also, we present the results of relation extraction in the form of a table. By comparing the F1 results of entity and relation subtasks, we intuitively see that our model also achieves good results in sub-tasks. And it proves that better entity recognition promotes the relation extraction effect from the side.

The dependency tree obtained through NLP tools depends on the quality of the sentence information and the correctness of the semantic. But in the real world, the text we need to process is not always shown as what we want. It is intricate and not standardised of the language expression. Therefore, we experiment to verify whether the construction of the adjacency matrix by its features is better than the result of the external dependency tree. Here, we use the NYT, WebNLG, and CoNLL04 datasets for comparative experiments. Because there are some incomprehensible sentences in the CoNLL04 dataset, it represents the problems that may exist in the text in reality. When the model structure is consistent, only the adjacency matrix method is changed from the attention layer to the dependency tree constructed by spaCy. We set Tree + as the method of using a dependency tree to get feature representation. We only change the way of obtaining the dependency tree, leaving the rest the same as MAGCN(ours). The experimental

TABLE 6 Comparative experimental results of using syntax tree and not using syntax tree

Method	Entity P	Entity R	Entity F1	Triples P	Triples R	Triples F1
CoNLL04						
Tree+	**83.3%**	69.3%	75.7%	**60.5%**	38.2%	46.9%
MAGCN(ours)	81.7%	**79.6%**	**80.7%**	58.3%	**50.8%**	**54.3%**
NYT						
Tree+	93.2%	94.4%	93.8%	80.7%	56.0%	66.1%
MAGCN(ours)	**93.8%**	**94.6%**	**94.2%**	**85.2%**	**59.7%**	**70.2%**
WebNLG						
Tree+	88.8%	94.6%	91.6%	81.9%	43.1%	56.4%
MAGCN (ours)	**93.4%**	**95.9%**	**94.7%**	**84.6%**	**50.3%**	**63.1%**

Note: These bolds values mean that the comparison result is the best.

Abbreviations: MAGCN, multi-head attention graph convolutional network model; NLP, Natural Language Processing.

results are shown in Table 6. For entity recognition, our model outperforms Tree + by 5%, performs better than Tree + by 7.4% at entity and relation triples predict on CoNLL04; outperforms Tree+ 0.4% in entities and 4.1% in relations triples on NYT; outperforms Tree+ 3.1% and 6.7% separately on WebNLG. These prove that it is better to construct adjacency matrix without relying on external tools.

As shown in Figure 7, we list the different relationships between entities on the NYT and WebNLG. Because there are so many relationships on NYT and WebNLG, we will only present some of them. By measuring the distance between the predicted relationships and the correct relationships category, we analyse and judge whether the use of external tools will impact the predicted results. The ordinate represents the distance, and the abscissa represents the control group used or not used external tools under datasets. The depth to the light colour of the heat map indicates the distance from the predicted result to the actual result. Compared with using external tools to generate dependency trees, most of the colours on the right were deeper than those on the left in three controlled groups. Our proposed model has a higher predictive effect on the vast majority of relation categories. We are more confident that our model has a higher discriminant ability in relation-distance recognition.

6 | CONCLUSION

The previous model relied on NLP tools to obtain the dependency tree, resulting in results that depend entirely on the accuracy of the extracted features. We present MAGCN, a joint extraction model based on multi-head attention and graph convolutional networks. Constructing the adjacency matrix through attention and combining with GCN to realise the joint extraction avoids the problem of external dependency analysis error. In addition, the relation extraction is expressed as a multi-head to solve the problem of overlapping relations. The problems of overlapping relation extraction and error analysis of external dependent tools in joint extraction are generally solved. We evaluate our model on several datasets. We have proved the accuracy of the proposed method through a large number of experiments and achieved 8.1% and 20.2% improvement on NYT and WebNLG datasets, respectively.

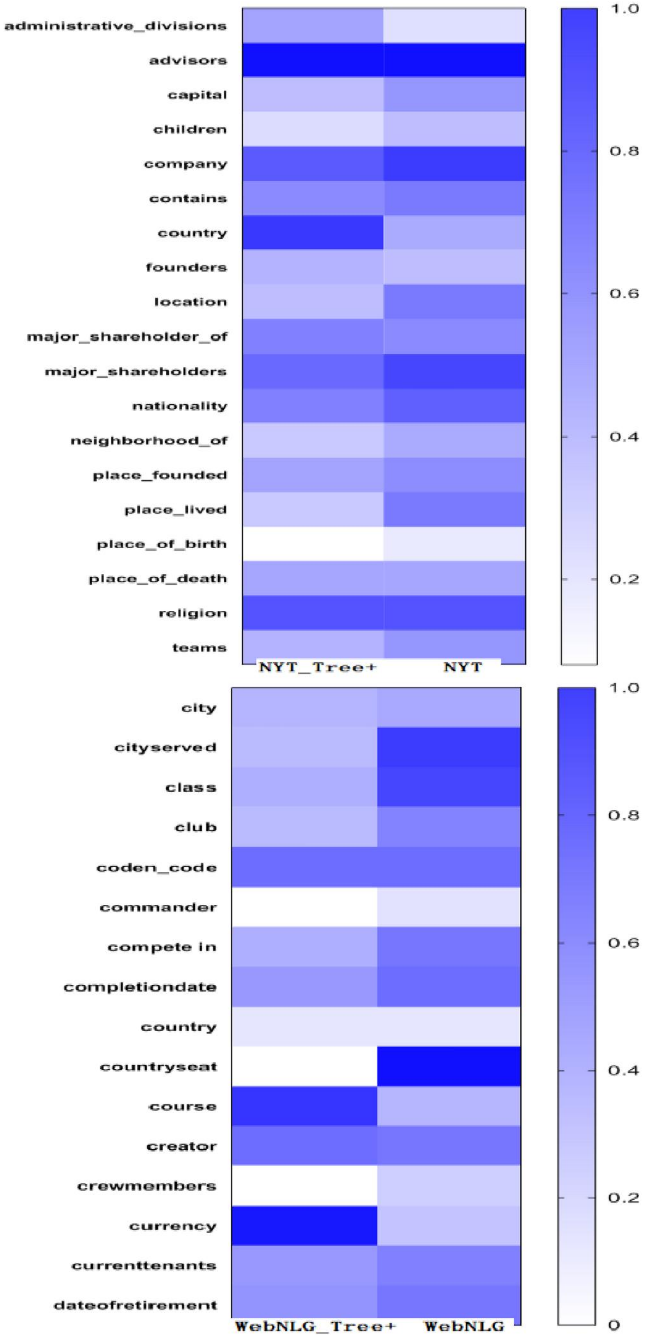

FIGURE 7 Visualisation of relationships between entities (above: NYT and below: WebNLG)

ACKNOWLEDGEMENTS

The State Key Program of National Natural Science of China, Grant/Award Number: 61533018; The National Natural Science Foundation of China, Grant/Award Number: 61402220; The Philosophy and Social Science Foundation of Hunan Province, Grant/Award Number: 16YBA323; Natural Science Foundation of Hunan Province, Grant/Award Number: 2020JJ4525; Scientific Research Fund of Hunan Provincial Education Department, Grant/Award Number: 18B279, 19A439.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Yongbin Liu https://orcid.org/0000-0002-3369-3101

REFERENCES

1. Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Meeting of Association of Computational Linguistics, pp. 178–181 (2004)
2. Fundel, K., Küffner, R., Zimmer, R.: Relation extraction using dependency parse trees. Bioinformatics. 23, 365–371 (2007)
3. Miwa, M., Sasaki, Y.: Modeling joint entity and relation extraction with table representation. In: Proceedings of EMNLP, pp. 1858–1869 (2014)
4. Qi, L., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of ACL, pp. 402–412 (2014)
5. Bekoulis, G., Johannes, D., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst. Appl. 114, 34–45 (2018)
6. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. Proc. ACL. (2017)
7. Liu, J., Gao, L., Guo, S.: A hybrid deep-learning approach for complex biochemical named entity recognition. Knowl. Based Syst. 221 (2020)
8. Wei, B., Hao, K., Gao, L., Tang, X.S.: Bio-inspired visual integrated model for multi-label classification of textile defect images. IEEE Trans. Cogn. Dev. Syst. 13(3), 503–513 (2020)
9. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp. 2537–2547. (2016)
10. Zhang, X., Cheng, J., Mirella, L.: Dependency parsing as head selection. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. (2017)
11. Guo, Z., Zhang, Y., Lu, W.: Attention guided graph convolutional networks for relation extraction. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. (2019)
12. Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. (2015)
13. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. (2017)
14. Bekoulis, G., Johannes, D., Demeester, T., Develder, C.: An attentive neural architecture for joint segmentation and parsing and its application to real estate ads. Expert Syst. Appl. 102, 100–112 (2018)
15. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investig. 30, 3–26 (2007)
16. Bach, N., Badaskar, S.: A review of relation extraction. Liter. Rev. Lang. Statist. II (2007)
17. Fu, T.-J., Li, P.-H., Ma, W.-Y.: GraphRel: modeling text as relational graphs for joint entity and relation extraction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. (2019)
18. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. (2017)
19. Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. (2018)
20. Fei, H., Renb, Y., Ji, D.: Boundaries and edges rethinking: an end-to-end neural model for overlapping entity relation extraction. Inf. Process. Manag. (2020)
21. Bai, C., Pan, L., Luo, S., Wu, Z.: Joint extraction of entities and relations by a novel end-to-end model with a double-pointer module. Neurocomputing (2020)
22. Katiyar, A., Cardie, C.: Going out on a limb: joint extraction of entity mentions and relations without dependency trees. In: Proceedings of the 55st Annual Meeting of the Association for Computational Linguistics. (2017)
23. Hochreiter, S., Jurgen Schmidhuber, J.: Long short-term memory. Neural Comput. 9, 1735–1780 (1997)
24. Kipf, T., Welling, M.: Semisupervised classification with graph convolutional networks. Proc. ICLR (2017)
25. Zhang, Y., Qi, P., Manning, C.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. (2018)
26. Riedel, S., Yao, L., McCallu, A.: Modeling relations and their mentions without labeled text. Proceedings of ECML-PKDD (2010)
27. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L., Roth, D., Wen, Y.: Creating training corpora for micro-planners: a linear programming formulation for global inference in natural language tasks. In: HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004), pp. 1–8. (2004)
28. Dan, R., Wen, Y. A linear programming formulation for global inference in natural language tasks. In HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004), 2004, pp. 1–8
29. Kingma, D., Jimmy, B.: Adam: a method for stochastic optimization. Proc. ICLR. (2015)

How to cite this article: Tao, Z., et al. Multi-head attention graph convolutional network model: End-to-end entity and relation joint extraction based on multi-head attention graph convolutional network. CAAI Trans. Intell. Technol. 8(2), 468–477 (2023). https://doi.org/10.1049/cit2.12086