End-to-end Relation-Enhanced Learnable Graph Self-attention Network for Knowledge Graphs Embedding

Shengchen Jiang
Kunming University of Science and Technology
Hongbin Wang (✉ whbin2007@126.com)
Kunming University of Science and Technology https://orcid.org/0000-0003-2176-2998
Xiang Hou
Chongqing University

Research Article

Keywords: knowledge graph embedding, relation-enhanced, convolutional knowledge subgraph, construction of negative samples

Posted Date: May 14th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-396932/v1

End-to-end Relation-Enhanced Learnable Graph Self-attention Network for Knowledge Graphs Embedding[♮]

Shengchen Jiang[‡], Hongbin Wang[†,‡] and Xiang Hou[§]

April 1, 2021

Abstract: The existing methods ignore the adverse effect of knowledge graph incompleteness on knowledge graph embedding. In addition, the complexity and large-scale of knowledge information hinder knowledge graph embedding performance of the classic graph convolutional network. In this paper, we analyzed the structural characteristics of knowledge graph and the imbalance of knowledge information. Complex knowledge information requires that the model should have better learnability, rather than linearly weighted qualitative constraints, so the method of end-to-end relation-enhanced learnable graph self-attention network for knowledge graphs embedding is proposed. Firstly, we construct the relation-enhanced adjacency matrix to consider the incompleteness of the knowledge graph. Secondly, the graph self-attention network is employed to obtain the global encoding and relevance ranking of entity node information. Thirdly, we propose the concept of convolutional knowledge subgraph, it is constructed according to the entity relevance ranking. Finally, we improve the training effect of the convKB model by changing the construction of negative samples to obtain a better reliability score in the decoder. The experimental results based on the data sets FB15k-237 and WN18RR show that the proposed method facilitates more comprehensive representation of knowledge information than the existing methods, in terms of Hits@10 and MRR.

Keywords: knowledge graph embedding; relation-enhanced; convolutional knowledge subgraph; construction of negative samples.

[b] This research was partially supported by the National Key R&D Program of China(Grant number:2018YFB1402900),the National Natural Science Foundation of China (Grant number: 61966020)and Chongqing Social Science Foundation(Grant number:2020YBTQ130).

[†] Corresponding author. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China. e-mail: whbin2007@126.com

[‡] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China.

[§] College of Automation, Chongqing University, Chongqing 400044, China.

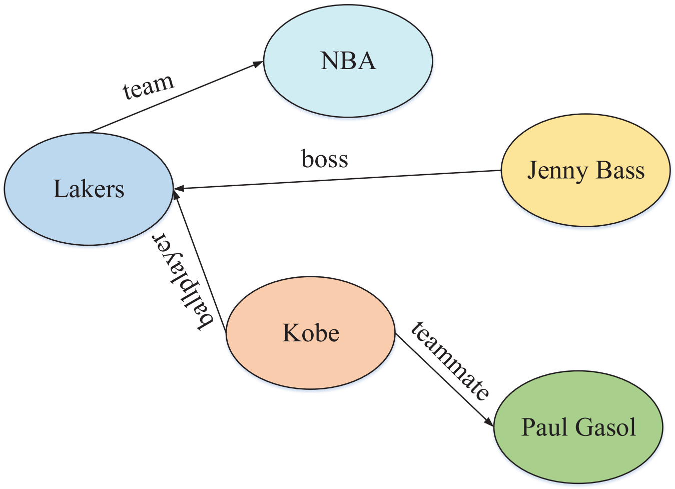

Figure 1: The local knowledge graph.

1 Introduction

The knowledge graph plays an indispensable role in the semantic network, and can benefit many downstream applications, such as personalized recommendation [1] and question answering [2]. knowledge information is represented by a network structure consisting of triples, in which nodes refers to entity information and connection edges represent multi-relation information among entities [3, 4], as shown in Fig.1. However, the symbolic of graph struction cannot be processed by machine learning, which leads to problem of insufficient utilization and analysis of knowledge base information. Therefore, the knowledge graph embedding plays an important role in the analysis of knowledge information, it aims to convert the knowledge information of the graph structure into a low-dimensional dense vector representation.

The knowledge graph is the semantic network graph describing various entities (or concepts) and relations in the real world. Each node in the graph represents an entity that is connected by various relation information to form different knowledge information. We analyzed knowledge information and general graph structure data, and there are obvious differences between knowledge graph and general graph structure data. The main points are as follows:

- Different knowledge information assigned by triples has different effects on entities. For example, the entity “Yao Ming” has different triples:(Yao Ming, Profession, basketball player) and (Yao Ming, wife, Ye Li). The knowledge of “Yao Ming” represents the global information related to it, but each entity will have a different impact on “Yao Ming”, of which “basketball players” have a greater impact.
- Knowledge graph connects diverse knowledge information through the complex relation

network. Compared with the simplified connection edges of general graph structure data, the degree of correlation between entities is much related to the relation category. However, even the same relation category has different effects on different entity information. For example, the "father-son" relation exerts different influence on entity "Qianlong" and "Yao Ming".

- The way of construction of knowledge graph determines the incompleteness of knowledge information, and the incompleteness of the knowledge graph will have a very terrible influence on the knowledge graph embedding.

To sum up, the complexity of such graphs demonstrates that the classical graph convolution network cannot handle knowledge information well.

In view of the above analysis, this paper proposes the end-to-end relation-enhanced learnable graph self-attention network for knowledge graphs embedding. First, we construct the relation-enhanced adjacency matrix. Secondly, the correlation between entity nodes is calculated by graph self-attention network, and the global encoding of the central entity node is obtained by weighted summation. And then, K neighboring nodes with the greatest correlation are selected as the knowledge subgraph, and the multi-layer convolution operation is embedded in the graph convolution process to obtain the entity encoding representation. Furthermore, the relational encoding representation is obtained by modeling the relational information, and the entity encoding and the relational encoding are combined to form the triple knowledge encoding. Finally, we improved the training effect of the convKB model by changing the construction of negative samples to obtain a better credibility score in the decoder.

The main contributions of the paper are presented in the following aspects:

1. We constructed a relation-enhanced learnable graph self-attention network, which is more in line with the complexity and diversity of knowledge information.
2. Based on the relevant ranking of entities, we put forward the concept of convolutional knowledge subgraphs, adding multi-layer convolutional networks into the aggregation process of graph convolutional networks, so that increase the learnability of the model.
3. We improved the training effect of the ConvKB model by changing the construction of negative samples to obtain a better credibility score in the decoder.

2 Related Work

The existing knowledge graphs embedding method is mainly based on the idea that the relationship is a transformation between entities. Bordes et al[5] proposed the TransE model based on the idea that the head entity could get the tail entity through the translation of the relationship information, which learned the embedding matrix of the entity and the relationship respectively. In subsequent studies, researchers proposed TransH [6], TransR [7], TransD [8], KG2E [9], TranSparse [10] by extending the embedding space of entities and relationships. Ebisu et al. changed the embedded space to sphere to obtain more comprehensive entity and relation embedding through spherical space [11]. Sun et al. defined the relationship as the rotation from head entity to tail entity, which increased the complexity of feature space in relation translation and obtained the knowledge graphs embedding in complex space [12]. However, these methods only model the characteristics of entities and relationships by constructing complex feature spaces, without considering the global structure information and incompleteness of the knowledge graph.

We focus more on recent research results on knowledge graphs embedding. The researchers used neural networks to capture more feature interactions between embeddings,respectively proposed the ConvE model [13], ConvKB model [14] and the InteractE model [15], which trained the knowledge embedding matrix through the credibility score function, and thus improves the expressiveness. Trouillon et al. deal with various binary relations through the combination of complex embedding [16]. Cai et al. used the generative adversarial network to improve the quality of negative samples and improve the learning effect of the model [17]. Recently, the graph convolutional networks are widely used in knowledge graph embedding, It has very well advantages for graph structure data [18, 19, 20, 21]. Shang et al. proposed to extend the ConvE model by using graph convolution network, maintaining the transformation characteristics between entities and relationships [22]. Many recent methods devote to preserving the symmetry and antisymmetry properties of relations to improve the expressiveness of embeddings [23, 24, 25, 26]. Chen et al. and Zhang et al. combined the bidirectional influence between entities and relationships by considering relationship edge information [27, 28]. Hamilton et al. embedded the logical query information of knowledge information into the knowledge graphs embedding by performing the logical operation in the low-dimensional embedded space [29]. However, the above methods do not take into account that compared with the general graph structure information, the knowledge information is imbalanced, so that the classic graph convolutional network cannot process the knowledge information well. In addition, the complexity of knowledge determines that models need better learnability, rather than linearly weighted qualitative constraints.

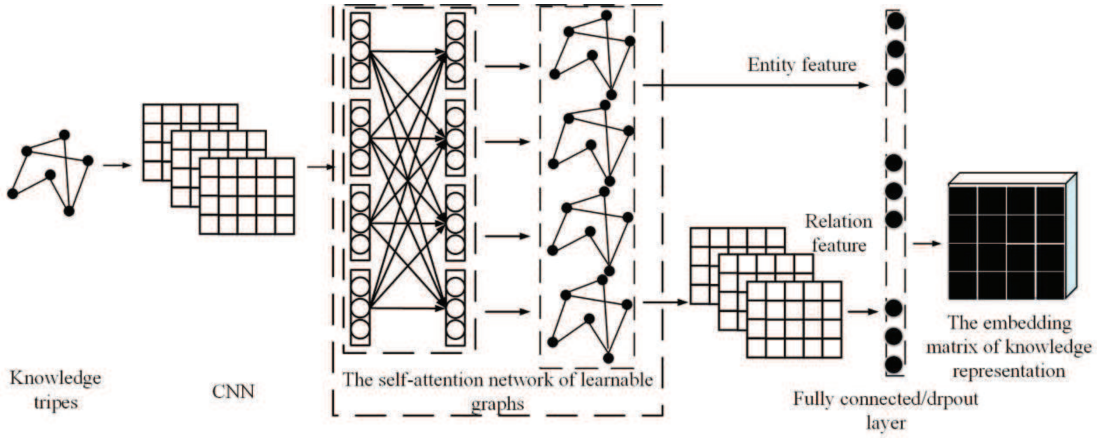

Figure 2: Knowledge encoder based on the relation-enhanced learnable graph self-attention network.

3 Our Approach

In this paper, we proposed the method of end-to-end relation-enhanced learnable graph self-attention network for knowledge graphs embedding. The method consists of two modules, which are knowledge encoder of relation-enhanced learnable graph self-attention network and the knowledge decoder of the reconstructing the negative sample.

3.1 Knowledge Encoder of the Relation-Enhanced Learnable Graph Self-attention Network

The knowledge graph is the semantic network graph describing various entities (or concepts) and relations in the real world. Each node in the graph represents an entity that is connected by various relation information to form different knowledge information. We analyzed the difference between knowledge information and general graph structure data. The interconnection and interaction of knowledge information caused each triple to affect the central entity differently. The complexity of knowledge graphs indicates that classical graph convolutional networks cannot handle knowledge information well. To this end, this paper proposes a knowledge information encoder that relation-enhanced learnable graph self-attention network, as shown in Fig.2.

The relation-enhanced learnable graph self-attention network is the extension of the classical GCN model. As Fig.2 shows, this model consists of multi-head self-attention layer and knowledge convolutional sublayer. It can assigns weights to each entity node according the influence degree of knowledge information, which is line with the characteristics of unequal knowledge information. The construction of convolutional knowledge subgraphs can increase the multi-level convolution operation in the process of graph aggregation. Compared with the classical graph convolution model, the feature extraction ability and learn ability of the model are improved. As well as, we construct the relation-enhanced adjacency matrix for the incompleteness of the knowledge graph. For each entity node in the graph, the entity node representation from the previous layer is used as input in the current layer network. The entity representation matrix of output is obtained through the relation-enhanced self-attention network of learnable graphs:

$$H_m^L = GKConv(AH^L W_{conv}) \tag{1}$$

Where A is relation-enhanced adjacency matrix of knowledge graph; $H^L \in R^{n \times F^L}$ is the entity representation matrix of the L layer learnable graph self-attention network, where F^L is the entity representation dimension; W_{conv} is the trainable parameter matrix

In order to avoid the adverse effect of incompleteness of knowledge graph on knowledge representation, we added indirect relationship attributes between entities in the adjacency matrix. We defined indirect relationship attributes including vertical indirect relationship and horizontal indirect relationship. The vertical indirect relationship means that the entities are in the same relationship path, as shown in Fig.3. There are triples in the knowledge base: (Kobe, players, Lakers), (Lakers, teams, NBA), the two triples are connected together through the relationship path, then ”Kobe” and ”NBA” have an indirect relationship, we call it is the vertical indirect relationship. The horizontal indirect relationship refers to the existence of cross entities between entities. As shown in Fig.3, there are triples in the knowledge base: (Kobe, players, Lakers), (Kobe, teammates, Gasol), then there is an indirect relationship between ”Gasol” and ”Lakers”, we call this it is the horizontal indirect relationship. The formula for calculating the indirect relationship index is:

$$r_{ij} = (\frac{1}{2})^{(k-1)} (\frac{1}{2})^p \tag{2}$$

Where k is the relationship path length; p is the horizontal indirect number. When there are multiple paths between two entities, take the shortest path. In this paper, we stipulate that there can only be one relationship between entity pairs. We do not calculate the indirect relationship attributes between directly adjacent entities. In this way, we construct the relation-enhanced adjacency matrix **A**:

$$\mathbf{A} = \begin{cases} 1 & direct \quad relationship \\ r_{ij} & indirect \quad relationship \\ 0 & no \quad relationship \end{cases} \tag{3}$$

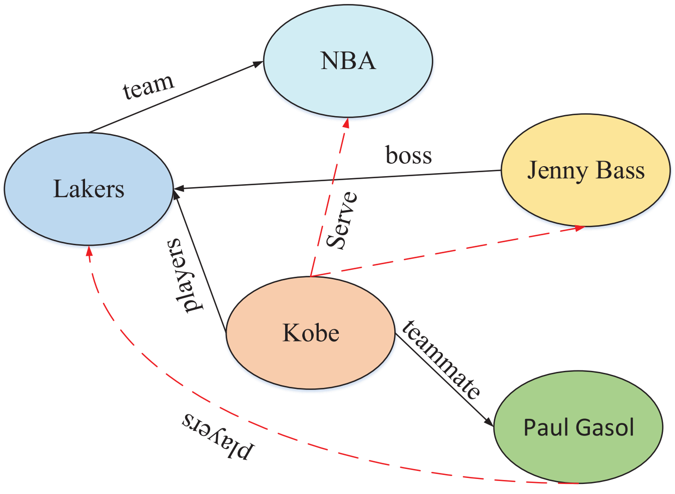

Figure 3: The indirect relationships of the knowledge graph.

For the relation-enhanced learnable graph self-attention network, as shown in Fig.2, the high-level feature representation of entities in the knowledge graph is extracted through the convolutional neural network:

$$H^1 = conv(e_1, e_2, \cdots, e_n) \tag{4}$$

Where $conv$ is the convolutional operation; e_i is the input characteristic of the $i - th$ entity; n is the entity number in the knowledge graph. Through the CNN model to obtain the input matrix of the entity representation H^1. In this paper, the feature representation of an entity consists of relationships and entities that are directly connected to the entity. It can be expressed as:

$$e_i = \{w_1^i, w_2^i, \cdots, w_m^i\} \tag{5}$$

Where w_j^i refers to the entity or relation directly adjacent to entity e_i. In this paper, m=50, (EMP is used to make up the deficiency). The classical graph convolutional neural network (GCN) assigns the same weight to the adjacent nodes, and obtains the representation of the central node by aggregating the feature representations of adjacent nodes. For the knowledge graph, the same relation category exerts different influences on different entity information. Through graph self-attention mechanism to obtain the attention weights between different entities:

$$e_{ij} = \sigma(h_i^L, h_j^L) \tag{6}$$

Where $h_i^L \in R^(F^L)$ and $h_j^L \in R^(F^L)$ are the feature representation of entities e_i and e_j at the layer L of the relation-enhanced learnable graphs self-attention network; σ function is the feedforward neural network of single layers with *LeakyReLU* as the activation function (negative input slope $\alpha = 0.2$). Attention weights between entity nodes are obtained through the normalization of scores calculated by single-layer feedforward neural network:

$$a_{ij} = \frac{exp(LeakyReLU(W_a^T[h_i^L \| h_j^L]))}{\sum_{g \in N_i} exp(LeakyReLU(W_a^T[h_i^L \| h_g^L]))} \tag{7}$$

Where $W_a^T \in R^{2F^L}$ is the weight matrix of feedforward neural network; $\|$ is the connection operation of vectors. At here form a relevance matrix $P \in R^{n \times n}$, $P_{ij} = a_{ij}$. Through aggregation of the neighboring entities to obtain the entity node representation of the fusion graph structure information:

$$h_i^{'} = \sigma(\sum_{g \in N_i} a_{ij} h_j^L) \tag{8}$$

Where N_i is the adjacency nodes set of entity e_i. In this paper, we constructed the relation-enhanced adjacency matrix **A** of knowledge graph, which reduces the negative effects of incomplete knowledge graph on knowledge representation. In the meanwhile, in view of the large amount of entity nodes in the graph, this paper makes references to Literature [30] with the multi-head self-attention mechanism, in order to acquire a better and more stable representation of entity. The final encoding representation of entity nodes is obtained by connection the feature vectors of each attention:

$$h_i = concate(\|_{k=1}^S \sigma(\sum_{g \in N_i} a_{ij}^k h_j^L)) \tag{9}$$

Where *concate* is the connection operation of vectors; S is the number of self-attention mechanism.

In the knowledge graph, the connection of each entity can reflect the semantic information of the entity. We hope to use convolutional neural networks to extract the connection information of each entity, but due to the different connection conditions of each entity, it is impossible to directly convolve the entity information in the knowledge graph. So we construct a convolutional knowledge subgraph, so that the convolution operation can be performed on the aggregated entity representation. As shown in Figure 4, In order of relevance, the convolutional knowledge subgraph of the entity ”Kobe” is [”Laker”, ”Gasol”, ”NBA”]. By performing the convolution operation on the convolutional knowledge subgraph, its characteristic representation is obtained.

we first extract the T entities with the most relevance for each entity through the correlation matrix **P** obtained, and construct a convolutional knowledge subgraph:

$$H_c = g(H, W, T) \tag{10}$$

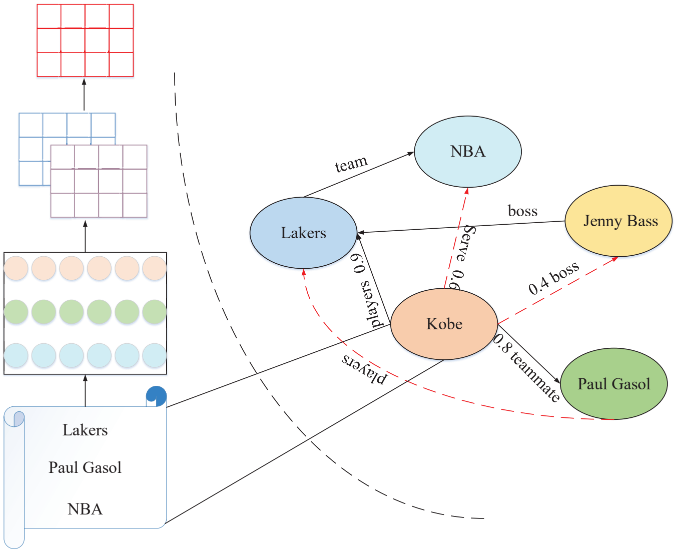

Figure 4: Construction of convolutional knowledge subgraph.

$$W = A \cdot P \tag{11}$$

$$P = \frac{\sum_{i \in S} P_i}{S} \tag{12}$$

Where g refers to the construction function of convolutional knowledge subgraph; **P** refers to the correlation obtained by the multi-head self-attention mechanism. In this paper, **P** is the average value of each self-attention mechanism; S is the number of self-attention mechanisms; T refers to the number of neighboring nodes with the greatest correlation, and if the neighboring node is less than T, then empty entity $< EMP >$ is adopted for supplement; **A** refers to the relation-enhanced adjacent matrix of the knowledge graph; Through matrix point multiplication to obtain the final relevance matrix **W** of adjacent nodes. And then obtain the entity representation through the convolutional neural network:

$$H^{L+1} = conv(W_1 H_c + b_1) \tag{13}$$

Where $conv$ is the convolutional operation; W_1 and b_1 are convolution layer parameters; $H_c \in R^{n \times T \times m}$ is the knowledge subgraph of entity. The convolution operation is performed on the convolutional knowledge subgraph to obtain an entity representation matrix $H^{L+1} \in R^{n \times F^{L+1}}$ as the output. In this process, we construct a convolutional knowledge subgraph, realizing the convolution operation in the graph convolution network, and reduce the model parameters. At the same time, the multi-layer convolution network can be embedded in the process of entity node information aggregation according to the graph structure, which improves the representation ability of the classic graph convolution network.

The knowledge graph embedding refers to the representation of entities and relationships. This paper holds the view that relations are manifested in the entity information. For example, with the father-son relation, people always think of Kangxi, Yongzheng and Yongzheng, Qianlong. Therefore, the relational embedding matrix is constructed with the idea that entity information can explain and reflect the relation information. First of all, the embedding matrix **H** of entities is employed to search the entity pairs corresponding to each relation category respectively. In this paper, 20 entities are selected as the information representation text according to each relationship category (EMP is used to make up the deficiency). The representation of each category is obtained by convolutional neural network:

$$R_i = conv(W_r X_i + b_r) \tag{14}$$

Where *conv* is the convolution operation; X_i is the information feature representation of the i-th relation category, composed of representations of entity pairs directly connected with relation categories (the entity representation of EMP is obtained by initialization); W_r and b_r are the parameter of convolutional neural network; R_i is the feature representation of the $i-th$ relation category. According to Equation 14, the relational embedding matrix **R** is obtained. Then the entity embedding matrix **H** is combined with the relation embedding matrix **R** to obtain the embedding matrix **U** of knowledge representation with the final encoding.

To sum up, the paper proposes a relation-enhanced learnable graph self-attention network model. The description of algorithm is expressed as follows Table 1.

3.2 Knowledge Decoder of the Improving Construction Negative Samples

Knowledge graph is a network structure composed of triples, and the decoder aims to define a reliability score function f, so that make the score of positive triples higher than the score of negative triples.

The embedding representation $U_i = [v_i^h, v_i^r, v_i^t] \in R^{3\times d}$ of knowledge information is obtained based on the relation-enhanced learnable graphs self-attention network, where v_h, v_r, v_t are the representations of head entity, relation and tail entity of knowledge triples and d is the dimension. Convolution operation is conducted for each dimension of the vector representation of knowledge triples:

$$v_i = conv(WU_{i,:} + b) \tag{15}$$

Where $U_{i,:} \in R^{3\times 1}$ is the $i-th$ dimensional representation of triple; convolutional kernel is $W \in R^(3\times1)$; conv function refers to the convolution operation, which is adopted to obtain the $i{-}th$

Table 1: Algorithm of relation-enhanced learnable graph self-attention network

Input: Input characteristic matrix of entity ($E = [e_1, e_2, \cdots, e_n]$), representation text of relational information (X), relation-enhanced adjacency matrix (A), size of knowledge subgraph (T), Number of self-attention mechanisms(S), Number of network layers (N).

Output: representation matrix of knowledge encoding(U)

```
Obtained entity representation matrix:
$H^1 \rightarrow H^1 = conv(e_1, e_2, \cdots, e_n)$.
For $L$ in range ($N$):
    For $k$ in range($S$):
        Calculate correlation weights between entities:
        $a_{ij}^k \rightarrow a_{ij}^k = \frac{exp(LeakyReLU(W_{a,k}^T[h_i^L \| h_j^L]))}{\sum_{g \in N_i} exp(LeakyReLU(W_{a,k}^T[h_i^L \| h_g^L]))}$
        Entity nodes are aggregated :
        $h'_{i,k} \rightarrow h'_{i,k} = \sigma(\sum_{g \in N_i} a_{ij}^k h_j^L)$
    Splicing of entity representation:
    $h_i \rightarrow h_i = concate(h'_{i,1}, h'_{i,2}, \cdots, h'_{i,s})$
    Constructing a convolutional knowledge subgraph:
    $H_c \rightarrow H_c = g(H, W, T)$
    Obtained entity output representation:
    $H^{L+1} \rightarrow H^{L+1} = conv(W_1 H_c + b_1)$
Obtained relation representation: $R_i \rightarrow R_i = conv(W_r X_i + b_r)$
Obtain the knowledge representation: $U \rightarrow U = [H^N, R]$
Return ($U$)
```

dimensional representation of triples. Perform repeated convolution of triples in each dimension. In order to maintain the conversion properties of the triplet, for the knowledge representation after the convolution, it is to choose the connection operation instead of reshaping:

$$v = [v_1, v_2, \cdots, v_d] \tag{16}$$

Where $v \in R^{1 \times d}$ is the feature representation of triples after convolution. Such connection operation not only extracts the global feature information of triples, but also maintains the transformation characteristics of triples. In order to acquire more abundant feature information, different convolution kernels are set to achieve the multi-channel convolution operation:

$$v_i^\tau = conv(W^\tau U_{i,:} + b^\tau) \tag{17}$$

Where W^τ and b^τ is the convolution layer parameters of the $\tau - th$ convolutional channel. The triple feature representation of each channel is connection to obtain the triple representation $V_i = [v^1, v^2, \cdots, v^t] \in R^{(}1 \times td)$, where t is the number of convolution channels. The reliability score of triple is calculated:

$$f(U_i) = \sigma(W_z V_i + b_z) \tag{18}$$

Where W_z and b_z is the full connection layer parameters. In this paper, we improved the training effect of negative samples on the convKB model by changing the construction of negative samples to obtain a better reliability score in the decoder. The negative samples are updated according to the reliability score after each iteration:

$$f(U_i^{'}) = \sigma(W_z V_i^{'} + b_z) \tag{19}$$

Where $U_i^{'}$ is a negative sample triple for U_i, the $U_i^{'} = (h^{'}, r, t^{'}) \in \left\{(h^{'}, r, t)|h^{'} \in \varepsilon, h^{'} \neq h\right\} \bigcup \left\{(h, r, t^{'})|t^{'} \in \varepsilon, t^{'} \neq t\right\}$ there refers to the entity set. With construction of negative samples, the random replacement method for head entity or tail entity is adopted in order to prevent the occurrence of simultaneous replacement of head entities and tail entities that are still positive samples. For the calculation of the negative sample score, the same weight matrix W_z and b_z is employed with no consideration of model loss and upgradation of model parameters. The triple with the highest reliability score is selected as the negative samples:

$$U_i^{'} = max(f(U_1^{'}), f(U_2^{'}), \cdots, f(U_q^{'})) \tag{20}$$

Where q is the number of constructed negative samples. The triple with the highest reliability score is chosen as the negative sample to participate in the next iteration training of the model. After each iteration, the negative samples are updated according to reliability score. With an aim to improve the calculation efficiency, randomly select 100 negative samples for each positive sample, and then the triple with the highest reliability score is used as the final negative sample. The model parameters are trained with the loss function:

$$\iota = \sum_{U \in \xi \bigcup \xi'} log(1 + exp(g(U)f(U))) + \frac{\lambda}{2}\|W\|_2^2 \tag{21}$$

$$\mathbf{A} = \begin{cases} -1 & U \in \xi \\ 1 & U^{'} \in \xi^{'} \end{cases} \tag{22}$$

Where ξ, $\xi^{'}$ refer to the set of positive and negative samples respectively. The parametric matrix W of the model is regularized by L2. Based on the optimization of loss function, the reliability score of positive triples are higher than that of negative triples. By optimizing the embedding matrix of knowledge representation for the relation-enhanced learnable graphs self-attention network, a higher quality knowledge embedding can be obtained.

4 Experiment

4.1 Experiment Data

This paper verification the effectiveness of the present approach based on two benchmark data sets: WN18RR [13] and FB15k-237 [15], which are the subsets of WN18 [5] and FB15k [5] respectively. The data set WN18RR consists of 40943 entities and 11 kinds of relation categories, the training set, validation set and test set contain 86835, 3034 and 3134 triples respectively. The data set FB15k-237 is composed of 14541 entities and 237 relation categories, there exist 272115, 17535 and 20466 triples in the training set, validation set and test set. The data sets WN18 and FB15k include many reversible relationships, which enable to predict triples easily. In the data sets of WN18RR and FB15k-237, the influence of reversible relationships be removed for a more authentic knowledge representation and the knowledge base completion.

4.2 Experimental Evaluation Criteria

In this paper, the experimental results of entity linking prediction are used to verify the effectiveness of the proposed method, and the prediction results are obtain according the reliability score function f on the triples of test set. During the testing process, this paper refers to the filtering protocol in Literature [5]. For each test triple, a group of negative triples is constructed through the random replacement of head entity and tail entity.

The present paper employs three benchmark evaluation indicators MR, MRR and Hits@10 to evaluate the effectiveness of the proposed method. The MR represents the average value of the correct label ranking in the probability distribution vector, and the smaller the value, the better. The MRR represents the average value of the reciprocal of the correct label ranking in the probability distribution vector, and the larger the value, the better. The Hits@10 represents the probability that the correct label rank in the top ten, and the larger the value, the better.

4.3 Setting of Experimental Parameters

In this paper, for experimental parameter settings, the output vector of convolution layer is set to be 64 dimensions; The number of convolution kernels is set to be 500 with the dropout of 0.6; The number of negative samples constructed by each positive sample in this paper is 100 and Select the one with the maximum reliability score as the negative sample; In the training process for end-to-end model, Adam optimizer is used to set the learning rate of 0.0001 and the parametric

Table 2: Experimental results of different sizes (T) of knowledge subgraphs

T	MR	MRR	Hits@10
10	346	0.341	52.5
20	323	0.353	54.9
30	304	0.367	55.6
40	272	0.376	56.1
50	311	0.370	56.3

regularization ratio is 0.001. The batch size set is 128 and the number of epoch is 800.

4.4 Experimental Results and Analysis

Experiment 1: In order to verify the effects of the size T (select the number of adjacent nodes with the greatest correlation) of convolutional knowledge subgraphs on the results of entity linking prediction, the size T of knowledge subgraphs at 10, 20, 30, 40 and 50 is tested on the result of entity linking prediction. Table 2 shows the effect of different sizes of knowledge subgraphs on the experimental results to verify the optimal size T of knowledge subgraphs:

As the experimental results in Table 2, the size of the knowledge subgraph increases, each evaluation indicators of the experimental results has been improved accordingly. According to the results, when T=40, achieve the best experimental results. One possible reason is that the size of the knowledge subgraph increases, more adjacent entity node information can be utilized, and convolution can obtain the richer feature representation of knowledge subgraph. But since our knowledge subgraphs are structured by relevance sort, when the knowledge subgraph reaches a certain level, the less relevant entity node information is likely to generate redundant information to the central entity, which will have a bad effect.

Experiment 2: In order to verify the effect of different numbers of self-attention mechanisms (h) on the results of entity linking prediction, the number of self-attention mechanisms (h) is selected to 1, 2, 4, 6, 8 and 10, so that comparative experiment of entity linking prediction can be performed. Table 3 below presents the effect of the number of different self-attention mechanism on the experimental results to verify the optimal number of self-attention mechanisms:

As the experimental results in table 3, the number (h) of attention mechanisms increases, each evaluation indicators of the experimental results has been improved accordingly. At the same time, according to the experimental results, whenh=10, achieve the best experimental results. One possible reason is that the number (h) of attention mechanisms increases, a better and more

Table 3: Comparative experimental results of different numbers (h) of self-attention mechanisms

h	MR	MRR	Hits@10
1	359	0.347	53.1
2	329	0.349	53.7
4	321	0.358	54.6
6	305	0.366	55.3
8	307	0.370	55.8
10	272	0.376	56.1

Table 4: Comparative experimental results of different methods

Method	MR	MRR	Hits@10
convKB [14]	257	0.396	51.7
GConKB	376	0.337	53.4
R-GATKB-N	272	0.376	56.1

stable of the features representation can be obtained.

Experiment 3: In order to verify the effect of adding knowledge encoder on the results of knowledge representation, this experiment adopts FB15k-237 data set and conducts a comparative analysis of convKB model, GCN + convKB model (GConKB) and relation-enhanced learnable graph self-attention network + convKB model (R-GATKB-N). Table 4 shows the experimental results with different methods to verify the effectiveness of end-to-end frameworks:

As the experimental results in table 4, the addition of knowledge encoder based on GCN, exerts minor influence on the knowledge representation quality. Only Hits@10 indicator has been improved, while MR and MRR indicators have decreased. The improvement of Hits@10 indicator shows the increasing number of high-quality knowledge representation, but the decrease of MRR indicators means some triples are poorly presented. In this regard, indication is made that the addition of knowledge encoding part and the consideration of graph struction information can enhance the quality of knowledge representation yet with less obvious effect. Compared with the classical GCN model, there is an apparent increase in terms of the indicators of MRR, Hits@10, etc. based on the relation-enhanced learnable graph self-attention network, among which Hits@10 increases by 4.4. Therefore, the model can effectively improve the quality of knowledge representation.

Experiment 4: In order to verify the effectiveness of the proposed method in knowledge representation, this experiment adopts FB15k-237 data set and selects 8 representative and highly influential methods from the related work in recent years as the objects for experimental compar-

Table 5: Comparison between the present and the baseline methods

Models	MR	MRR	Hits@10
TransE [5]	347	0.294	46.5
DISTMULT [31]	254	0.241	41.9
COMPLEX [16]	339	0.247	42.8
KBGAN [17]	-	0.278	45.8
ConvE [13]	246	0.316	49.1
ConvKB [14]	257	0.396	51.7
RotatE [12]	177	0.338	53.3
SACN [22]	-	0.350	54.0
R-GAT	324	0.357	54.4
R-GATKB	298	0.373	55.6
R-GATKB-N	272	0.376	56.1

ison. The experimental results are presented in Table 5:

All the baseline results in Table 5 are copied from the original paper, and the missing data indicate that there is no corresponding report score in the original paper. Here, R-GAT is the experimental result of adding the relation-enhanced adjacency matrix. R-GATKB is the experimental result of adding the relation-enhanced adjacency matrix and the convolution knowledge subgraph. R-GATKB-N is the experimental result of adding the relation-enhanced adjacency matrix and the convolution knowledge subgraph and improving the negative samples. The experimental results show that drawing on the relation-enhanced learnable graphs self-attention network, takes into full consideration the more comprehensive information of knowledge graphs and improves the learn ability of the model for a better effect of the model. By comparison with the SACN [22], Obvious increases are shown in Hits@10 and MRR, which shows that the based on the graph structure and relation-enhanced adjacency matrix, considering the imbalance of knowledge information and the construction of convolutional knowledge subgraphs can increase the learn ability of model, so that obtain a better knowledge representation.

Experiment 5: In this experiment, In order to test the applicability on different data sets, this method is implemented on the data set WN18RR, and the applicability is verified by the effect therefrom. The experimental results are shown in Table 6.

The experimental results in Table 6 indicate the method proposed in this paper improves the evaluation indexes Hits@10 and MRR in the data set WN18RR compared with the latest research results of SACN [22]. Furthermore, the applicability of the proposed method can be verified

Table 6: Comparative experimental results of different methods in the WN18RR

Models	MR	MRR	Hits@10
TransE [5]	3384	0.226	50.1
DISTMULT [31]	5110	0.430	49.0
COMPLEX [16]	5261	0.440	51.0
KBGAN [17]	-	0.213	48.1
ConvE [13]	5277	0.460	48.0
ConvKB [14]	2554	0.248	52.5
RotatE [12]	3340	0.476	57.1
SACN [22]	-	0.470	54.0
R- GATKB-N	3169	0.488	55.4

owing to the better results achieved on different data sets in knowledge representation.

4.5 Conclusions

Aiming at the complexity and large-scale of knowledge information determine that the classic graph convolutional network cannot achieve better knowledge graph embedding. The paper puts forward the method of end-to-end relation-enhanced learnable graphs self -attention network for knowledge graphs embedding. The proposed method promotes the flexibility and learn ability of network model. The entity linking prediction experiments on public data sets have achieved good results. In the near future, great emphasis should be put on the construction of knowledge-driven neural network model, which transforms the feature learning of model into the knowledge learning with a higher level.

5 Acknowledgements

This work was supported by the National Key R&D Program of China (2018YFB1402900), the Nature and Science Foundation of China (61966020) and Chongqing Social Science Foundation(Grant number:2020YBTQ130). The authors are grateful to the referee for his helpful comments and constructive suggestions which have contributed to the final preparation of this paper.

6 Declarations

Conflict of interest The authors declare that they have no conflict of interest.
Human and animal rights This article does not contain any studies with human or animal subjects performed by any of the authors.
Informed consent Informed consent was not required as no human or animals were involved.
Authorship contributions Hongbin Wang puts forward the idea of the thesis and guides the students to realize the code. Shengchen Jiang participated in the programming and writing of the thesis. Xiang Hou participated in the revision and improvement of the thesis.

References

[1] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In Proceedings of WWW. 1835C1844 (2018)

[2] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li: Knowledge graph embedding based question answering. In Proceedings of WSDM. 105C113 (2019)

[3] Alberto, et al: Combining two and three-way embedding models for link prediction in knowledge bases. Journal of Artificial Intelligence Research. 55,715-742 (2016)

[4] Nickel, Maximilian, et al: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE.104(1),11-33 (2015)

[5] Bordes, Antoine, et al: Translating embeddings for modeling multi-relational data. presented at the Advances in neural information processing systems. 2787-2795 (2013)

[6] Wang Z, Zhang J, Feng J, et al: Knowledge graph embedding by translating on hyperplanes. Twenty-Eighth AAAI conference on artificial intelligence. 1532-1543 (2014)

[7] Lin Y, Liu Z, Sun M, et al: Learning entity and relation embeddings for knowledge graph completion. Twenty-ninth AAAI conference on artificial intelligence. (2015)

[8] Ji, Guoliang, et al: Knowledge graph embedding via dynamic mapping matrix. presented at the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. 687-696 (2015)

[9] He S, Liu K, Ji G, et al: Learning to represent knowledge graphs with gaussian embedding. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM. 623-632 (2015)

[10] Ji G, Liu K, He S, et al: Knowledge graph completion with adaptive sparse transfer matrix. Thirtieth AAAI Conference on Artificial Intelligence. (2016)

[11] Ebisu T, Ichise R. Toruse: Knowledge graph embedding on a lie group. Thirty-Second AAAI Conference on Artificial Intelligence. (2018)

[12] Sun Z, Deng Z H, Nie J Y, et al: RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. arXiv preprint arXiv:1902.10197. (2019)

[13] Dettmers T, Minervini P, Stenetorp P, et al: Convolutional 2d knowledge graphembeddings. Thirty-Second AAAI Conference on Artificial Intelligence. (2018)

[14] Nguyen D Q, Nguyen T D, Nguyen D Q, et al: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121. (2017)

[15] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar: Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. arXiv preprint arXiv:1911.00219. (2019)

[16] Trouillon T, Welbl J, Riedel S, et al: Complex embeddings for simple link prediction. International Conference on Machine Learning. 2071-2080 (2016)

[17] Cai L, Wang W Y. Kbgan: Adversarial learning for knowledge graph embeddings. NAACL. (2018)

[18] Schlichtkrull M, Kipf T N, Bloem P, et al: Modeling relational data with graph convolutional networks. European Semantic Web Conference. Springer, Cham. 593-607 (2018)

[19] Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P.: A Convolutional Neural Network for Modelling Sentences. In Proceedings of ACL 2014. 1, 655C665, (2014)

[20] Defferrard, M.; Bresson, X.; and Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Proceedings of NIPS. 3837C3845 (2016)

[21] Schlichtkrull M, Kipf T N, Bloem P, et al: Modeling relational data with graph convolutional networks. European Semantic Web Conference. Springer, Cham. 593-607 (2018)

[22] Shang C, Tang Y, Huang J, et al: End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion. arXiv preprint arXiv:1811.04441. (2018)

[23] Nathani D, Chauhan J, Sharma C, et al: Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 4710-4723 (2019)

[24] Boyang Ding, Quan Wang, Bin Wang, and Li Guo: Improving knowledge graph embedding using simple constraints. In Proceedings of ACL. 110C121 (2018)

[25] Seyed Mehran Kazemi and David Poole: Simple embedding for link prediction in knowledge graphs. In Proceedings of NIPS. 4289C4300 (2018)

[26] Canran Xu and Ruijiang Li: Relation embedding with dihedral group in knowledge graph. In Proceedings of ACL. 263C272 (2019)

[27] Chen H, Sun X, Tian Y, et al: Enhanced network embeddings via exploiting edge label-s. Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 1579-1582 (2018)

[28] Zhang W, Paudel B, Zhang W, et al: Interaction Embeddings for Prediction and Explanation in Knowledge Graphs. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. 96-104 (2019)

[29] Hamilton W, Bajaj P, Zitnik M, et al: Embedding logical queries on knowledge graphs. Advances in Neural Information Processing Systems. 2026-2037 (2018)

[30] Vaswani, Ashish, et al: Attention is all you need. presented at the Advances in neural information processing systems. 5998-6008 (2017)

[31] Yang B, Yih W, He X, et al: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575. (2014)