HIGH TECHNOLOGY LETTERS | Vol. 27 No. 1 | Mar. 2021 | pp. 43-52

doi:10.3772/j. issn. 1006-6748.2021.01.006

Joint learning based on multi-shaped filters for knowledge graph completion①

Li Shaojie(李少杰)②* **, Chen Shudong**, Ouyang Xiaoye**, Gong Lichen* **
(* School of Microelectronics, University of Chinese Academy of Sciences, Beijing 100049, P. R. China)
(** Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, P. R. China)

Abstract

To solve the problem of missing many valid triples in knowledge graphs (KGs), a novel model based on a convolutional neural network (CNN) called ConvKG is proposed, which employs a joint learning strategy for knowledge graph completion (KGC). Related research work has shown the superiority of convolutional neural networks (CNNs) in extracting semantic features of triple embeddings. However, these researches use only one single-shaped filter and fail to extract semantic features of different granularity. To solve this problem, ConvKG exploits multi-shaped filters to co-convolute on the triple embeddings, joint learning semantic features of different granularity. Different shaped filters cover different sizes on the triple embeddings and capture pairwise interactions of different granularity among triple elements. Experimental results confirm the strength of joint learning, and compared with state-of-the-art CNN-based KGC models, ConvKG achieves the better mean rank (MR) and Hits@10 metrics on dataset WN18RR, and the better MR on dataset FB15k-237.

Key words: knowledge graph embedding (KGE), knowledge graph completion (KGC), convolutional neural network (CNN), joint learning, multi-shaped filter

0 Introduction

Knowledge graphs (KGs), such as YAGO[1], DBpedia[2], Freebase[3], are graph-structured knowledge bases, where each edge called a fact or a triple is represented in the form of (*headentity*, *relation*, *tailentity*), or (h, r, t) in short. These KGs have numerous structured information which is a useful resource for many natural language processing tasks, such as question answering[4] and machine reading[5]. However, KGs typically are incomplete, missing a lot of valid triples[6-7]. This problem has attracted much attention and gives rise to knowledge graph completion or link prediction, which refers to predict whether a triple (h, r, t) is valid or not[8]. State-of-the-art link prediction solutions are primarily knowledge graph embedding (KGE) based models. KGE aims to project both entities and relations into a continuous low-dimensional space. These distributed representation vectors, called entity embeddings or relation embeddings, can efficiently measure the semantic correlations between entities and relations, which can significantly benefit a variety of downstream tasks such as knowledge graph completion (KGC) and knowledge inference. KGE models can be roughly classified as four groups: translational distance models[8-18], tensor product models[19-22], neural network models[23-25] and convolutional neural network (CNN)[26-28] based models. Translational distance models and tensor product models tend to adopt shallow, simple encoders to extract latent features, thus only capture the linear relationships between entities. Neural network models often have more parameters and are prone to overfit[21]. Recent research[26-28] has raised interest in applying convolutional neural networks for KGC and proved the superiority of convolutional neural networks in generating more expressive triple embeddings. CNN based models can extract the complex semantic features between entities and the relation in a triple due to modeling the non-linear relationship and the efficiency of parameter utilization. However, existing CNN-based models exploit only one single-shaped filter, failing to capture diverse triple features. For example, ConvE[26] uses 2D filters to extract the semantic features of the matrices concatenated by head entities and relations. ConvKB[27] uses 1D filters to capture the transitional characteristics of each dimension on the triple embeddings. ConvKE[28] uses 2D filters to extract the semantic features of the matrices concatenated by entities and relations.

① Supported by the National Natural Science Foundation of China (No. 61876144).
② To whom correspondence should be addressed. E-mail: chenshudong@ime.ac.cn
Received on Mar. 30, 2020

In this paper, a novel convolutional neural network embedding model called ConvKG is proposed, which uses the joint learning strategy based on multi-shaped filters to extract the semantic features on triples embeddings, capturing pairwise interactions of different granularity among triple elements. Specifically, ConvKG explores three different shaped filters: 1×3, 3×3, and 5×3, sliding on the triple embedding row by row respectively. The intuition is that 1×3 shaped filters capture fine-grained semantic features, 3×3 shaped filters capture intermediate granularity semantic features, and 5×3 shaped filters capture coarse-grained semantic features due to covering bigger sizes on the triple embeddings. After getting diverse semantic features, ConvKG uses the method of weight allocation to let the model learn what granularities of semantic features are needed. Evaluating the performance of ConvKG on two benchmark datasets WN18RR[26] and FB15k-237[29], the experimental results show that, compared with state-of-the-art CNN-based KGC models, ConvKG obtains better mean rank (MR) and Hits@10 metrics on dataset WN18RR, and better MR on dataset FB15k-237. Experimental results also show that ConvKG performs better than major translational distance models and tensor product models.

Contributions in this work are as follows.

A novel model called ConvKG is introduced, which has exquisite structure and is effective in downstream tasks like knowledge graph completion.

A joint learning strategy based on multi-shaped filters for knowledge graph completion is proposed, extracting semantic features of different granularity on triple embeddings.

By using knowledge graph completion task, the performance of ConvKG on two benchmark datasets WN18RR and Hits@10 is evaluated.

1 Related work

KGC is a crucial task for knowledge graphs. A good variety of models have been proposed for this task. These models can be introduced into four categories: translational distance models, tensor product models, neural network models, and CNN-based models.

Translational distance models measure the plausibility of a triple as the distance between the two entities, usually after a translation operated by the relation. TransE[9] is the most representative translational distance model. TransE assumes that the embedding of the head entity plus the embedding of the relation should be closed to the embedding of the tail entity. Since TransE cannot effectively model the three complex relations of 1-to-N, N-to-1, and N-to-N[10], a series of TransE-based improved models such as TransH, TransR and TransD are proposed. TransH[10] introduces relation-specific hyperplanes, using translation vectors and hyperplane normal vectors to represent relations. And TransR[11] introduces relation-specific spaces, using different matrices to represent different relations. However, TransD[12] believes that the head entities and tail entities should have different relation matrices, so two different relation matrices are used to represent the relation in a triple, but inevitably introducing too many parameters. For solving this problem, TransD exploits two vectors to construct each relation matrix. There are other translation-based models, such as TransM[14], TransF[15], TransA[16], STransE[13], UM[17], SE[8], and TranSparse[18]. Although translational models have fewer parameters and use simple operations, however, they have difficulties in learning expressive triple embeddings due to using linear operators.

Ref. [33] unifies tensor product models[19-22] and neural network models[23-25] under the framework of semantic matching models. Semantic matching models measure the plausibility of a triple as a degree of latent semantic match between entities and relation by a similarity-based scoring function. RESCAL[19] represents each entity as a vector and each relation as a matrix, then a bilinear function is exploited to model pairwise interactions among triple embedding factors. DistMult[22] is an extension of RESCAL by simplifying relation matrices to diagonal matrices, so this model captures pairwise interactions along the same dimension of the triple embeddings and reduces the number of parameters of relations. Complex[20] extends DistMult by introducing complex-valued embeddings to better model asymmetric relations. And HolE[21] represents relation as a vector and explores circular operations to model pairwise interactions between the head entity and the tail entity, which creates more efficient and scalable triple embeddings. Tensor product models use simple multiplication operators, thus only capture the linear relationships between entities. Despite neural network models exploit complex encoders, e. g., NTN[23], MLP[24], and NAM[25], this kind of approach often has more parameters and are prone to overfit[21].

There are three major models based on convolutional neural networks for knowledge graph completion[26-28]. ConvE[26] reshapes the head entity vector and the relation vector in a triple and then concatenates them into an input matrix, which is performed with multiple 2D filters to obtain different feature maps. After concatenating these extracted feature maps, the outputting matrix is vectorized and mapped into the same vector space as the tail entity, then a dot operation is applied in the two vectors and the result is a scale which indicates the score of a triple. ConvKB[27] represents each triple as a 3-column matrix where each column represents a triple element. Then multiple 1D filters are operated on the matrix to get different feature maps which are then concatenated and multiplied with a weight vector to output a score which entails the plausibility of the triple. Specifically, ConvKB learns the global relationships among same dimensional entries of a triple due to using 1D filters. ConvKE[28] adopts the dimension transformation strategy to improve the sliding steps of the convolution sliding window on the triple embeddings and the information interaction ability of entities and relations in more dimensions. ConvKE also uses 2D convolution sliding windows to enhance the receptive field to capture the whole information in more dimensions of triples. Experimental results of the three models suggest CNN performs better than tensor product models and translation-based models, and CNN-based models can learn more expressive KG embeddings due to its capturing complex relationships by learning non-linear semantic features with efficient parameters.

But ConvKB, ConvKE, and ConvE take into account only one-single shaped filters and fail to extract diverse semantic features. Specifically, ConvE[26] and ConvKE[28] use 2D filters to extract the semantic features of the embedding matrices, and ConvKB[27] uses 1D filters to capture the transitional characteristics of each dimension on the triple embeddings, as discussed in the introduction part. Instead, ConvKG exploits a joint learning strategy based on multiple shaped filters to co-convolute on the triple embeddings, which can generate semantic feature maps of different granularity. The framework of ConvKG is introduced in detail in Section 2.

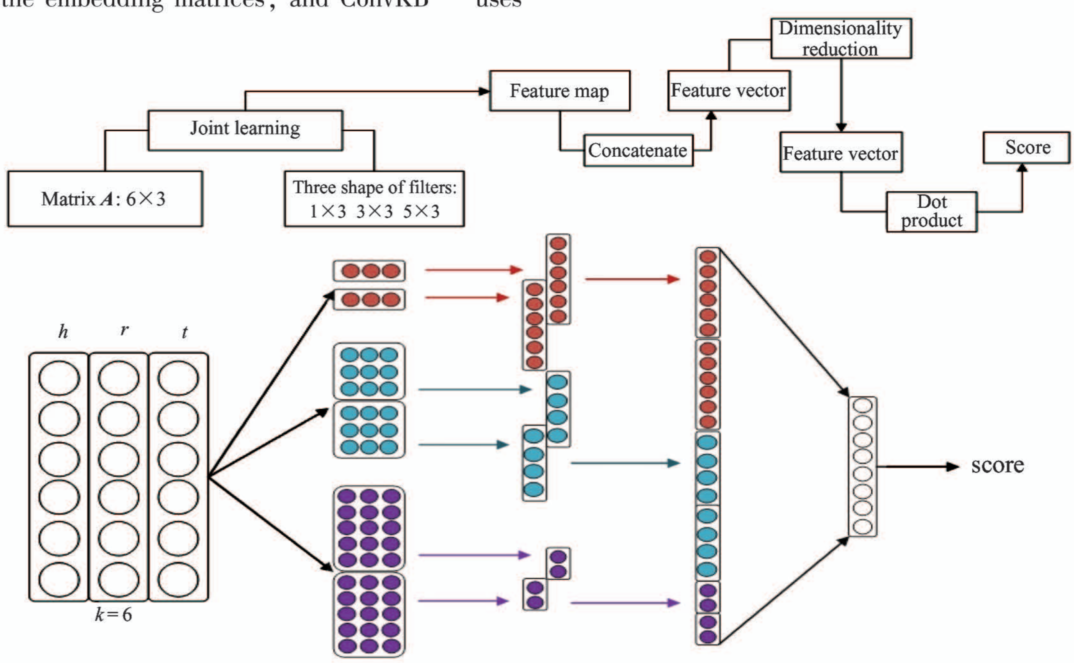
2 The proposed approach

2.1 Background

In this section, notations and definitions used in the rest of the paper are introduced firstly. A knowledge graph is denoted by $\boldsymbol{G} = (\boldsymbol{E}, \boldsymbol{R})$, where $\boldsymbol{E}$ and $\boldsymbol{R}$ represent the set of entities and relations, respectively. A triple (h, r, t) is represented as an edge r between two nodes h and t in $\boldsymbol{G}$. This framework aims to learn a score function $f(h, r, t)$ which gives a higher score for a valid triple.

2.2 Framework

ConvKG can be divided into two steps. The first step is to use the filters with different granularity to extract the features of the triple embeddings. The second step is the weight allocation, to let the model learn what granularities of semantic features are needed. The framework of ConvKG is summarized in Fig. 1. As shown in Fig. 1, ConvKG first exploits the joint learning strategy



(Suppose the embedding size $k=6$, the number of filters $N=2$; ConvKG first exploits joint learning strategy based on three shaped filters to output feature maps with different semantic granularity on the matrix $\boldsymbol{A}$, then concatenates these feature maps to obtain a feature vector and output a score after a dimensionality reduction operation and a dot product operation)

Fig. 1 Illustrations of ConvKG

based on three shaped filters to output feature maps with different semantic granularity on the matrix $\boldsymbol{A}$, after concatenates these feature maps to obtain a feature vector, ConvKG then uses weight allocation to choose the semantic features with important granularity and output a score which indicates the plausibility of the triple.

The convolutional neural network was mainly applied in image processing tasks[34-37]. It imitates human visual experience, uses filters with local perception to capture local detailed information of the image, and then adds convolutional layers to capture more complex, more abstract, and higher-level feature information of images, and finally outputs abstract representations on different dimensions of images. ConvKG regards each triple as an image and uses a convolutional neural network to extract its features. Specifically, ConvKG distributes triples as vectors in k-dimensional vector space in the form of $(\boldsymbol{h}^k, \boldsymbol{r}^k, \boldsymbol{t}^k)$ and concatenates them, constructing a matrix $\boldsymbol{A} = [\boldsymbol{h}^k, \boldsymbol{r}^k, \boldsymbol{t}^k] \in \boldsymbol{R}^{k\times 3}$ which represents an image of a triple. ConvKG explores three different shaped filters: 1×3, 3×3, and 5×3. Specifically, the filters $\boldsymbol{\omega}_1 \in \boldsymbol{R}^{1\times 3}$, $\boldsymbol{\omega}_2 \in \boldsymbol{R}^{3\times 3}$, and $\boldsymbol{\omega}_3 \in \boldsymbol{R}^{5\times 3}$ slid on the triple embedding matrix $\boldsymbol{A}$ row by row, generating three shaped feature maps $\boldsymbol{m}_1 \in \boldsymbol{R}^{k\times 1}$, $\boldsymbol{m}_2 \in \boldsymbol{R}^{(k-2)\times 1}$, $\boldsymbol{m}_3 \in \boldsymbol{R}^{(k-4)\times 1}$ respectively, where $\boldsymbol{m}_1 \in \boldsymbol{R}^{k\times 1}$ represents the extracting fine-grained semantic features outputted by $\boldsymbol{\omega}_1 \in \boldsymbol{R}^{1\times 3}$, $\boldsymbol{m}_2 \in \boldsymbol{R}^{(k-2)\times 1}$ represents the extracting intermediate-granularity semantic features outputted by $\boldsymbol{\omega}_2 \in \boldsymbol{R}^{3\times 3}$ and $\boldsymbol{m}_3 \in \boldsymbol{R}^{(k-4)\times 1}$ represents the extracting coarse-grained granularity semantic features outputted by $\boldsymbol{\omega}_3 \in \boldsymbol{R}^{5\times 3}$. These feature maps are outputted by the following formula: $\boldsymbol{m}_i = \boldsymbol{g}(\boldsymbol{\omega}_i * \boldsymbol{A}) + \boldsymbol{b}$, where $*$ is a convolution operation, $\boldsymbol{b}$ is a bias term, $\boldsymbol{g}$ is some activation function such as Relu, sigmoid. What's more, the 1D filters $\boldsymbol{\omega}_1 \in \boldsymbol{R}^{1\times 3}$ can capture the global translational characteristics among same dimensional entries of a triple[17], and the 2D filters $\boldsymbol{\omega}_2 \in \boldsymbol{R}^{3\times 3}$ and $\boldsymbol{\omega}_3 \in \boldsymbol{R}^{5\times 3}$ can capture the more information interactions due to the larger receptive fields[26,28]. Different shaped filters cover different sizes on the triple embeddings and capture pairwise interactions of different granularity among triple elements. These different shaped feature maps extracted from matrix $\boldsymbol{A}$ are used to improve the diversities of the extracting embedding triple features, to produce richer, more complex and more expressive feature maps.

After getting feature maps with different granularity, ConvKG uses the method of weight allocation to let the model learn what granularities of semantic features are needed. Specifically, suppose each set of different shaped filters has N number of filters, ConvKG gets a whole filter set Ω, i. e. $|\Omega| = 3N$. The filters $\boldsymbol{\omega}_1 \in \boldsymbol{R}^{1\times 3}$ generate N number of feature maps $\boldsymbol{m}_1 \in \boldsymbol{R}^{k\times 1}$, the filters $\boldsymbol{\omega}_2 \in \boldsymbol{R}^{3\times 3}$ generate N number of feature maps $\boldsymbol{m}_2 \in \boldsymbol{R}^{(k-2)\times 1}$, and the filters $\boldsymbol{\omega}_3 \in \boldsymbol{R}^{5\times 3}$ generate N number of feature maps $\boldsymbol{m}_3 \in \boldsymbol{R}^{(k-4)\times 1}$. ConvKG concatenates all these feature maps, obtaining a vector $\boldsymbol{v}_{feature} \in \boldsymbol{R}^{3N(3k-6)\times 1}$ which is then computed with a weight matrix $\boldsymbol{w}_1 \in \boldsymbol{R}^{7k\times 3N(3k-6)}$ via a linear transformation. Linear transformation chooses important vector elements by assigning different weights, which are updated by backpropagation, and output a vector $\boldsymbol{v}_{trans} \in \boldsymbol{R}^{7k\times 1}$. $\boldsymbol{v}_{trans}$ is then computed by a non-linear operation. The formula is defined as follows: $\boldsymbol{v}_{trans} = \boldsymbol{g}(\boldsymbol{v}_{feature} \times \boldsymbol{w}_1)$, where $\times$ is a matrix multiplication operation, $\boldsymbol{g}$ is some activation function such as Relu, sigmoid. The vector outputted by the above operations is then matched with a weight vector $\boldsymbol{w}_2 \in \boldsymbol{R}^{7k\times 1}$ via a dot product operation, outputting a score which indicates the plausibility of a triple $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$.

Formally, the score function of ConvKG is defined as

$$f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) = \boldsymbol{g}(vec(\boldsymbol{g}(\boldsymbol{A} * \Omega)) \times \boldsymbol{w}_1) \cdot \boldsymbol{w}_2 \tag{1}$$

where Ω, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are shared parameters which are independent of triples; $*$ is a convolution operation; $\times$ is matrix multiplication operation.

For training its parameters, ConvKG uses Adam optimizer[30] and minimizes the following logistic loss function ζ with L_2 regulation on the weight parameter $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$\zeta = \sum_{(h,r,t)\in\{D\cup D'\}} \log(1+\exp(\boldsymbol{l}_{(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t})} \cdot f(\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}))) + \lambda(\|\boldsymbol{w}_1\|_2^2 + \|\boldsymbol{w}_2\|_2^2) \tag{2}$$

where $l_{(h,r,t)} = \begin{cases} -1 & \text{for } (h,r,t) \in D \\ 1 & \text{for } (h,r,t) \in D' \end{cases}$, where D' is a set of invalid triples generated by corrupting valid triples in D. This loss function aims to obtain a score function, that will give a higher score for a valid triple and a lower score for an invalid triple. And the L_2 regulation aims to avoid the overfitting of the model on the training dataset.

3 Experiments

3.1 Datasets

Two benchmark datasets are chosen to evaluate the performance of ConvKG: FB15K-237 and WN18RR. FB15K-237 is a subset of FB15K which is a subset of Freebase where most triples are related to movies and sports. And WN18RR is a subset of WN18 which is a subset of WorldNet where most triples consist of hyponym and hypernym relations. Ref. [29] mentioned that the two datasets FB15K and WN18 have a large number of test triples which can simply be obtained by inverting training triples. Ref. [26] introduced a simple rule-based model that can achieve state-of-the-art results on both WN18 and FB15k. Therefore, corresponding subset datasets FB15k-237 and WN18RR were created to avoid this reversible problem in WN18 and FB15k. In this paper, these two more convincing datasets are used to evaluate the performance of ConvKG. Table 1 provides the statistics of WN18RR and FB15k-237.

Table 1 Statistics of the experimental datasets

Datasets	Entities (N_e)	Relations (N_r)	Train set	Valid set	Test set
WN18RR	40 943	11	86 836	3034	3134
RB15k-237	14 541	237	272 115	17 535	20 466

3.2 Training protocol

Invalid triples D' are created by corrupting valid triples D. Specifically, a negative fact can be generated by replacing either head entity h or tail entity t with a random sample from valid triples D, i. e.

$$D' = \{(h', r, t) \mid h' \in E \wedge h' \neq h \wedge (h, r, t) \in D\} \cup \{(h, r, t') \mid t' \in E \wedge t' \neq t \wedge (h, r, t) \in D\} \quad (3)$$

And entity and relation embeddings are initialized with TransE[9] model. Especially on the WN18RR dataset, ConvKG exploits triple embeddings produced by the TransE model which is trained by the pre-trained 100-dimensional Glove word embeddings. And the hyperparameters of TransE are set as follows[20]: embedding size $k = 100$, learning rate at $5e^{-4}$, margin $r = 1$, and L1-norm for FB15k-237; embedding size $k = 100$, learning rate at $5e^{-4}$, margin $r = 5$, and L1-norm for WN18RR. And ConvKG explores three shaped filters $\in \{1 \times 3, 3 \times 3, 5 \times 3\}$ initialized by three truncated normal distribution to jointly learn triple embeddings. ConvKG selects the number of each shaped filter in a range of $\{50, 100, 200\}$ and finally chooses 200 for WN18RR and 100 for FB15k-237.

The rest parameters of ConvKG include weight matrix $\boldsymbol{w}_1$ and weight vector $\boldsymbol{w}_2$. Adam is used to optimizing all these parameters with initial learning rate $5e^{-5}$ and $\lambda = 1e^{-5}$ for WN18RR and initial learning rate $1e^{-5}$ and $\lambda = 1e^{-7}$ for FB15k-237. ConvKG sets the batch size as 256 and uses ReLU as the activation function. By training ConvKG up to 100 epochs, parameters saved on the last epoch are used for the evaluation.

3.3 Evaluation protocol

KG completion or link prediction task aims at predicting a missing entity when given a relation and another entity, i. e., predict h given (r, t) or predict t given (h, r).

Following previous work[9], a set of corrupt triples for each valid test triple (h, r, t) are generated by replacing either h or t with every other entity $e_i' \in E - e_i$. And ConvKG evaluates its performance in a filtered setting, i. e. taking no account of any corrupt triples which are already present in the KG. During the evaluation, ConvKG assigns scores to the valid test triple and its corresponding corrupted triples, then ranks these scores in ascending order and gets the rank of the valid test triple. Following Ref. [9], two common evaluation metrics are used: mean rank (MR), and Hits@10 (i. e., the proportion of the correct triples in the top 10 predictions). Lower MR or higher Hits@10 represents better performance.

3.4 Results and analysis

Table 2 shows the prediction results of different models on the test sets of the two datasets WN18RR and FB15k-237. The results show that ConvKG achieves the best MR on two benchmark datasets and the highest Hits@10 on WN18RR than previous state-of-the-art CNN-based KGC models. Specifically, ConvKG gains improvements of $391 - 204 = 187$ in MR (which is 47+% relative improvement) and $59 - 52.5 = 6.5\%$ in Hits@10 for WN18RR dataset and also gains improvements of $205 - 164 = 41$ (which is 20% relative improvement) in MR for FB15k-237 dataset.

ConvKG is first compared with ConvKE(NDT)[28]. ConvKE(NDT) refers to take no dimension transformation strategy, and the difference between it and ConvKG is that ConvKE(NDT) only uses the 3×3 shaped filters to extract semantic feature on triple embeddings, whereas ConvKG also uses 1×3 and 5×3 shaped filters except for using 3×3 shaped filters. And ConvKG is then compared with ConvKB. ConvKB uses 1×3 shaped filters to extract semantic feature on triple em beddings, whereas ConvKG also uses 3×3 and 5×3 shaped filters except for using 1×3 shaped filters. From

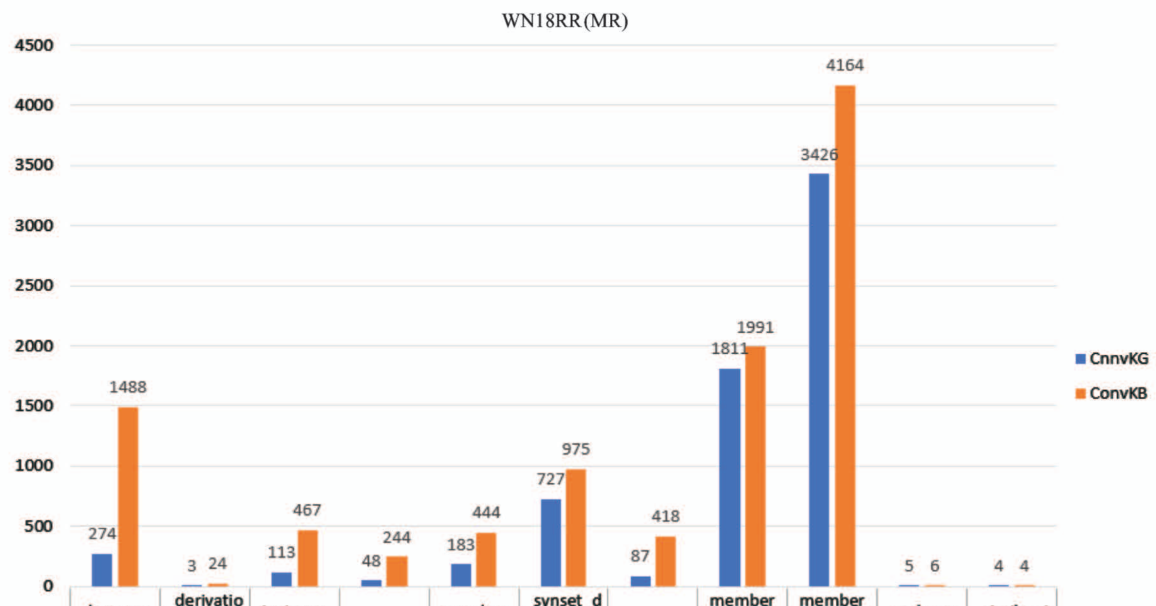
Table 2 Experimental results of different models on WN18RR and FB15k-237 test sets (The best score is in bold and the second score is underlined)

	WN18RR		FB15k-237	
	MR	Hits@ 10	MR	Hits@ 10
Distmult[22]	5110	49	254	41.9
ComplEx[20]	5261	51	339	42.8
TransE[9]	3384	50.1	347	46.5
R-GCN[32]	6700	8	600	30
ConvE[26]	5277	48	246	<u>49.1</u>
ConvKB[27]	2554	<u>52.5</u>	257	**51.7**
ConvKE(NDT)[28]	562	48.8	210	47.1
ConvKE[28]	391	50	205	45.1
This work	**204**	**59**	**164**	45.8

the above analysis, it can be seen that ConvKG focuses on extracting semantic feature maps with different granularity, whereas ConvKB and ConvKE(NDT) focus on extracting semantic features of single granularity on triple embeddings. From Table 2, compared with ConvKG and ConvKE(NDT), it can be seen that ConvKG achieves the better MR on two benchmark datasets and the higher Hits@ 10 on WN18RR, indicating that the joint learning strategy based on multi-shaped filters is beneficial in generating expressive embeddings due to its capturing pairwise interactions of different granularity among triple elements.

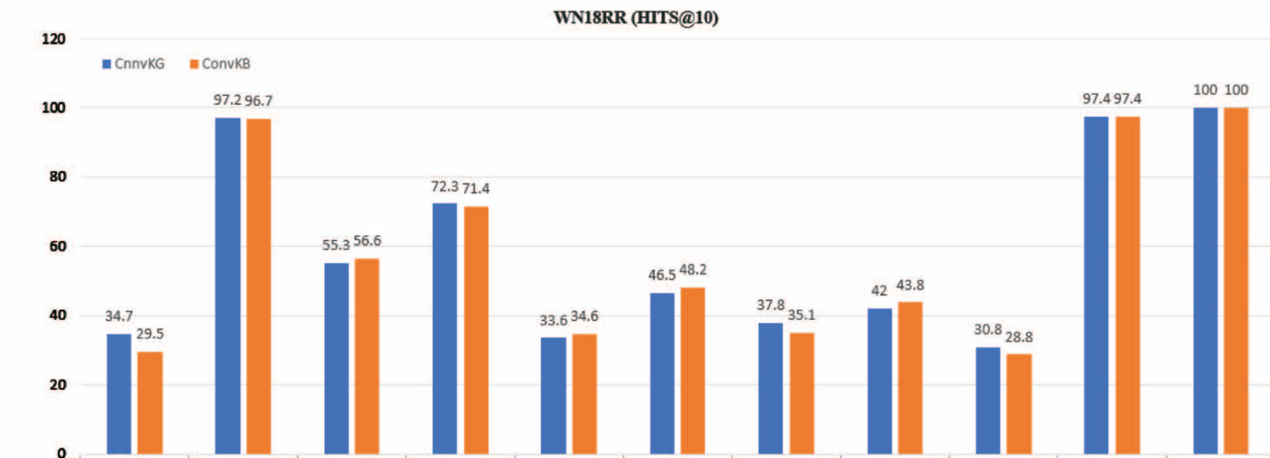
Following Ref. [9], on two benchmark datasets, ConvKG is further compared with ConvKB on the predicting results w. r. t each relation or each relation category. ConvKG employs the truncated normal distributions to initialize filters to extract semantic features of different granularity, and ConvKB exploits [0.1, 0.1, −0.1] which is a TransE-based way to initialize filters to capture the global translational characteristics among same dimensional entries of a triple[27].

For the WN18RR test set, ConvKG is compared with ConvKB on the predicting results w. r. t each relation since its rare number of relations and see *also _ see*, *similar _ to*, *verb _ group* and *derivationally _ related _ form* as M-M relations. Fig. 2 illustrates that ConvKG performs better than ConvKB on MR metric in each relation. Fig. 3 illustrates that ConvKG performs



	_hypernym	_derivationally_related_form	_instance_hypernym	_also_see	_member_meronym	_synset_domain_topic_of	_has_part	_member_of_domain_usage	_member_of_domain_region	_verb_group	_similar_to
CnnvKG	274	3	113	48	183	727	87	1811	3426	5	4
ConvKB	1488	24	467	244	444	975	418	1991	4164	6	4

Fig. 2 MR on WN18RR test set w. r. t each relation

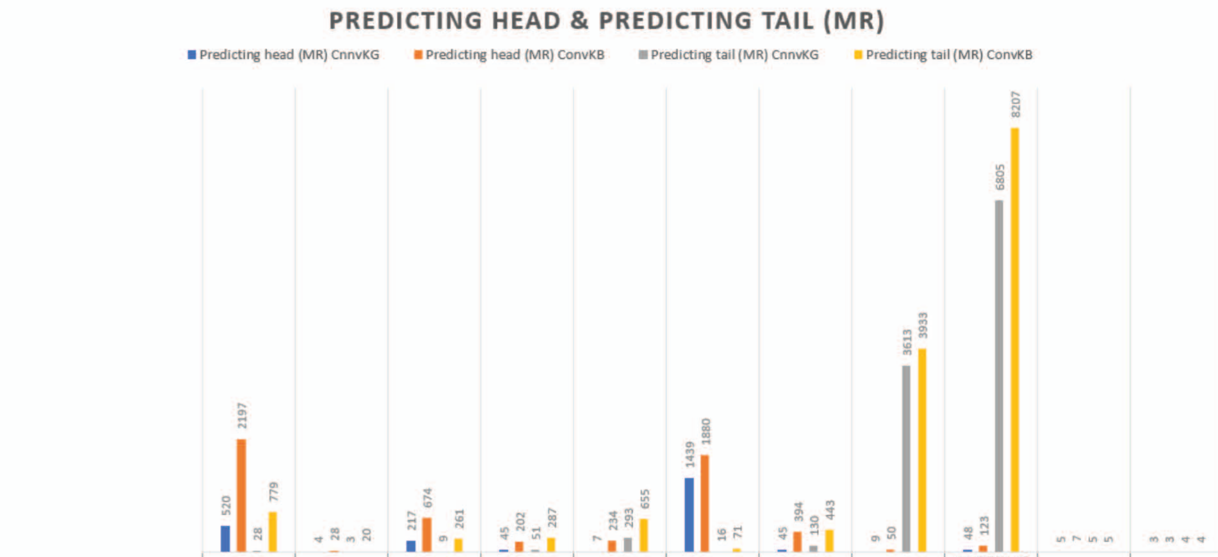


	_hypernym	_derivationally_related_form	_instance_hypernym	_also_see	_member_meronym	_synset_domain_topic_of	_has_part	_member_of_domain_usage	_member_of_domain_region	_verb_group	_similar_to
CnnvKG	34.7	97.2	55.3	72.3	33.6	46.5	37.8	42	30.8	97.4	100
ConvKB	29.5	96.7	56.6	71.4	34.6	48.2	35.1	43.8	28.8	97.4	100

Fig. 3 Hits@ 10 on WN18RR test set w. r. t each relation

better than ConvKB on Hits@ 10 metric in M-M relations. And for *hypernym* and *derivationally _ related _ form* relations which account for 40% and 34% of WN18RR test set respectively, ConvKG also has an improvement compared to ConvKB on Hits@ 10 metric. Fig. 4 shows that ConvKG performs better than ConvKB on MR in each relation whether in predicting head entities or predicting tail entities. Fig. 5 shows that ConvKG obtains the better Hits@ 10 in M-M relations or in *hypernym* and *derivationally _ related _ form* relations which take a large proportion of WN18RR test set whether in predicting head entities or predicting tail entities. After the conclusion of the above experimental results, it is beneficial to adopt joint learning based on multi-shaped filters for knowledge graph completion on WN18RR.

For the FB15k-237 test set, the average number α of head entities per tail entity and the average number β of tail entities per head entity are counted. r is categorized 1-1 relation if $\alpha < 1.5$ and $\beta < 1.5$, 1-*M* relation if $\alpha < 1.5$ and $\beta \geqslant 1.5$, *M*-1 relation if $\alpha \geqslant 1.5$ and $\beta < 1.5$ or *M*-*M* relation if $\alpha \geqslant 1.5$ and $\beta \geqslant 1.5$. Results



	_hypernym	_derivationally_related_form	_instance_hypernym	_also_see	_member_meronym	_synset_domain_topic_of	_has_part	_member_of_domain_usage	_member_of_domain_region	_verb_group	_similar_to
Predicting head (MR) CnnvKG	520	4	217	45	7	1439	45	9	48	5	3
Predicting head (MR) ConvKB	2197	28	674	202	234	1880	394	50	123	7	3
Predicting tail (MR) CnnvKG	28	3	9	51	293	16	130	3613	6805	5	4
Predicting tail (MR) ConvKB	779	20	261	287	655	71	443	3933	8207	5	4

Fig. 4 MR of predicting head and tail entities on WN18RR test set w. r. t each relation

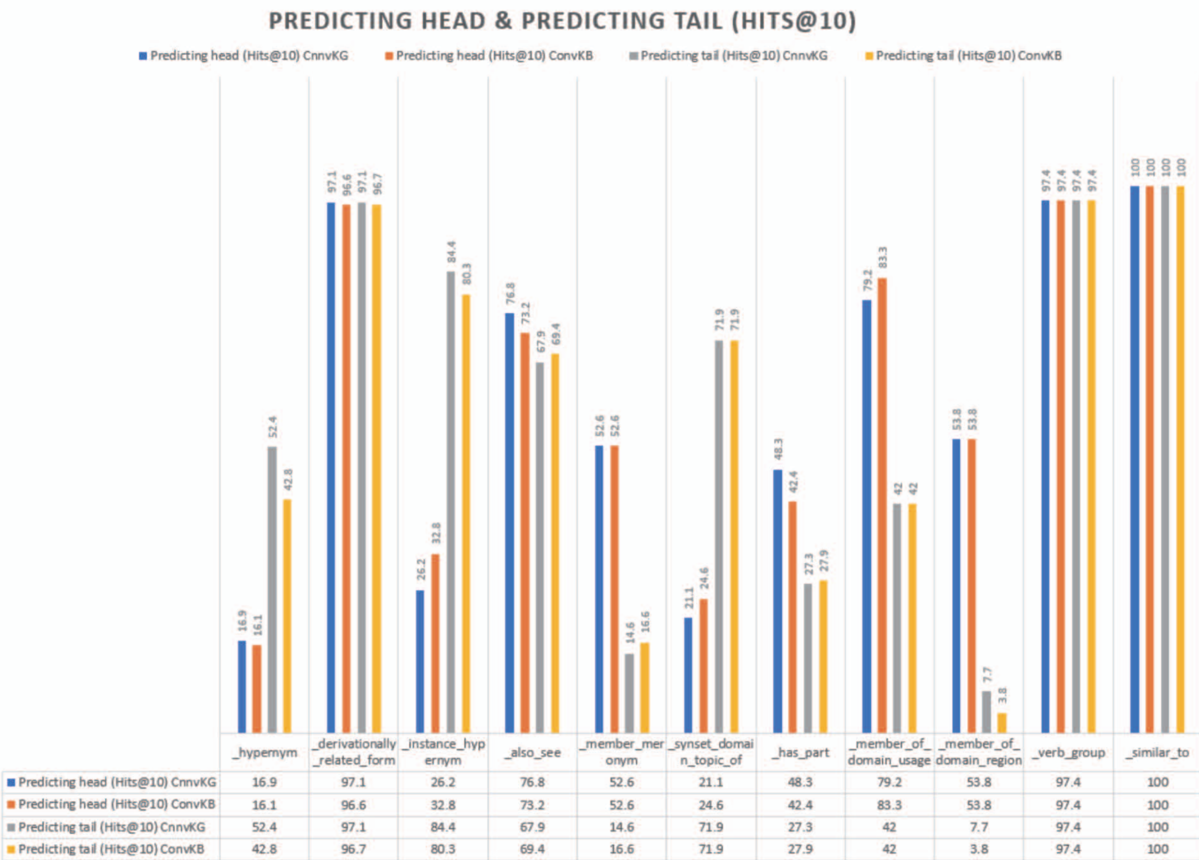


Fig. 5 MR of predicting head and tail entities on WN18RR test set w. r. t each relation

show that 17, 26, 86 and 108 relations are assigned to 1-1, 1-M, M-1 and M-M relations, respectively. Fig. 6 shows that, on the FB15k-237 test set, ConvKG performs better than ConvKB on MR metric in all relation categories. Although ConvKG has poor performance on Hits@ 10 metric in most relation categories than ConvKB as shown in Fig. 7, it is worth noting that ConvKG performs better when compared to ConvKE which is a KGC model based on convolutional neural

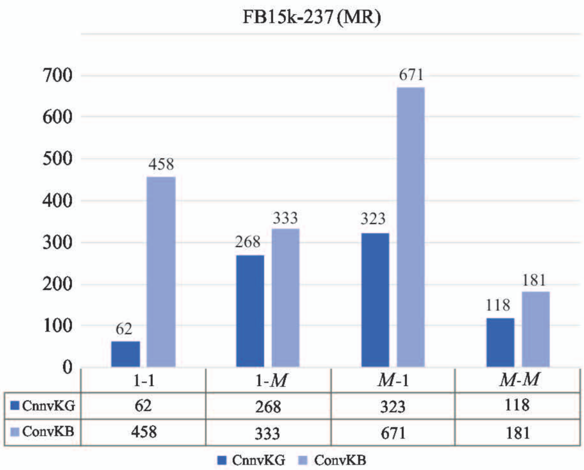


Fig. 6 MR on FB15k-237 test set w. r. t each relation category

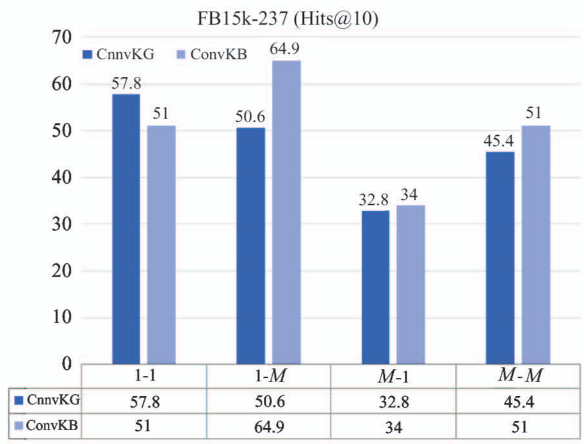


Fig. 7 Hits@ 10on FB15k-237 test set w. r. t each relation category

network. From the above analysis on FB15k-237 dataset, it can be concluded that the joint learning strategy based on multi-shaped filters can improve the performance of the MR metric on the FB15k-237 dataset.

Table 2 also shows that models based on CNNs perform better than tensor product models and translation-based models in KG completion, which indicates that CNN works well in modeling complex relationships and produces more expressive embeddings. Through the above analysis on two benchmark datasets, it can be concluded that joint learning based on multi-shaped filters can improve the performance for KG completion due to its capturing pairwise interactions of different granularity among triple elements, generating richer and more expressive embeddings. Especially for WN18RR dataset, the joint learning strategy performs well because it can capture pairwise interactions with different receptive fields among triple elements and extract semantic features of different granularity. But for FB15k-237 dataset, compared with other CNN-based KGC models, joint learning strategy is less effective on Hits@ 10 metric due to bringing noise semantic features, but it also achieves the best MR.

4 Conclusions

In this work, ConvKG, which is a knowledge graph embedding model based on convolutional neural networks for KG completion, is proposed. And for extracting semantic feature maps of different granularity, ConvKG employs a joint learning strategy based on multi-shaped filters co-convoluting on the triple embeddings, generating more diverse and expressive triple embeddings. ConvKG uses two benchmark datasets FB15k-237 and WN18RR to evaluate its performance, and experimental results show that, compared with state-of-the-art CNN-based KGC models, ConvKG achieves the better MR on two benchmark datasets and a higher Hits@ 10 on WN18RR, indicating that the joint learning strategy can improve the performance of KG completion. But ConvKG models the triples separately and ignores the relationship between triples. In fact, the entities of each triple contain rich neighbor information. In future, the relation paths will be considered to model the neighbor information.

References

[1] Suchanek F M, Kasneci G, Weikum G. YAGO: a core of semantic knowledge[C] // Proceedings of the 16th International Conference on World Wide Web, Banff, Alberta, Canada, 2007: 697-706

[2] Lehmann J, Isele R, Jakob M, et al. DBpedia-A large-scale, multilingual knowledge base extracted from Wikipedia[J]. *Social Work*, 2015, 6(2): 167-195

[3] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C] // Proceedings of the ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, 2008: 1247-1250

[4] Zhang Y, Liu K, He S, et al. Question answering over knowledge base with neural attention combining global knowledge information[EB/OL]. https://arxiv.org/pdf/1606.00979v1.pdf: arXiv, 2016

[5] Yang B, Mitchell T. Leveraging knowledge bases in LSTMs for improving machine reading[C] // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017: 1436-1446

[6] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[C] // Advances in Neural Information Processing Systems, Nevada, USA, 2013: 926-934

[7] West R, Gabrilovich E, Murphy K, et al. Knowledge base completion via search-based question answering[C] //The 23rd International Conference on World Wide Web the Web Conference, Seoul, Korea, 2014: 515-526

[8] Bordes A, Weston J, Collobert R, et al. Learning structured embeddings of knowledge bases[C] //AAAI Conference on Artificial Intelligence, San Francisco, USA, 2011: 301-306

[9] Bordes A, Usunier N, Garciaduran A, et al. Translating embeddings for modeling multi-relational data[C] // Advances in Neural Information Processing Systems, Nevada, USA, 2013: 2787-2795

[10] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes [C] // The 28th AAAI Conference on Artificial Intelligence, Quebec City, Canada, 2014: 1112-1119

[11] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion [C] // The 29th AAAI Conference on Artificial Intelligence, Austin, USA, 2015: 2181-2187

[12] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix[C] //Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing, Beijing, China, 2015: 687-696

[13] Nguyen D Q, Sirts K, Qu L, et al. STransE: a novel embedding model of entities and relationships in knowledge bases[C] //North American Chapter of the Association for Computational Linguistics, San Diego, USA, 2016: 460-466

[14] Fan M, Zhou Q, Chang E, et al. Transition-based knowledge graph embedding with relational mapping properties [C] // Pacific Asia Conference on Language Information and Computation, Phuket, Thailand, 2014: 328-337

[15] Feng J, Huang M, Wang M, et al. Knowledge graph embedding by flexible translation[C] //Principles of Knowledge Representation and Reasoning, Cape Town, South Africa, 2016: 557-560

[16] Xiao H, Huang M, Hao Y, et al. TransA: an adaptive approach for knowledge graph embedding[EB/OL]. https://arxiv.org/pdf/1509.05490.pdf: arXiv, 2015

[17] Bordes A, Glorot X, Weston J, et al. Joint learning of words and meaning representations for open-text semantic parsing[C] //International Conference on Artificial Intelligence and Statistics, Cádiz, Spain, 2012: 127-135

[18] Ji G, Liu K, He S, et al. Knowledge graph completion with adaptive sparse transfer matrix [C] // National Conference on Artificial Intelligence, Phoenix, USA, 2016: 985-991

[19] Nickel M, Tresp V, Kriegel H P. A three-way model for collective learning on multi-relational data[C] //The 28th

International Conference on Machine Learning, Bellevue, Washington, USA, 2011: 809-816

[20] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C] // International Conference on Machine Learning, New York, USA, 2016: 2071-2080

[21] Nickel M, Rosasco L, Poggio T, et al. Holographic embeddings of knowledge graphs[EB/OL]. https://arxiv.org/pdf/1510.04935v2.pdf: arXiv, 2015

[22] Yang B, Yih W, He X, et al. Embedding entities and relations for learning and inference in knowledge bases [EB/OL]. https://arxiv.org/abs/1412.6575: arXiv, 2014

[23] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[C] // Neural Information Processing Systems, Nevada, USA, 2013: 926-934

[24] Dong X, Gabrilovich E, Heitz G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C] // Knowledge Discovery and Data Mining, New York, USA, 2014: 601-610

[25] Liu Q, Jiang H, Ling Z, et al. Probabilistic reasoning via deep learning: neural association models[EB/OL]. https://arxiv.org/pdf/1603.07704.pdf: arXiv, 2016

[26] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D knowledge graph embeddings [EB/OL]. https://arxiv.org/abs/1707.01476: arXiv, 2017

[27] Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A novel embedding model for knowledge base completion based on convolutional neural network[C] // North American Chapter of the Association for Computational Linguistics, New Orleans, USA, 2018: 327-333

[28] Li S, Chen S, Hao Y, et al. Efficient knowledge representation model based on convolutional neural network [J]. *Chinese High Technology Letters*, 2020, 30(9): 901-907 (In Chinese)

[29] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference [C] // The 3rd Workshop on Continuous Vector Space Models and their Compositionality, Beijing, China, 2015: 57-66

[30] Kingma D P, Ba J. Adam: a method for stochastic optimization [EB/OL]. https://arxiv.org/pdf/1412.6980.pdf: arXiv, 2014

[31] Nguyen D Q, Vu T, Nguyen T D, et al. A capsule network-based embedding model for knowledge graph completion and search personalization[C] // North American Chapter of the Association for Computational Linguistics, Minneapolis, USA, 2019: 2180-2189

[32] Schlichtkrull M S, Kipf T, Bloem P, et al. Modeling relational data with graph convolutional networks[C] // European Semantic Web Conference, Heraklion, Greece, 2018: 593-607

[33] Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: a survey of approaches and applications [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(12): 2724-2743

[34] Fu X, Li L, Mao K, et al. Remote sensing image classification based on convolution neural network model[J]. *Chinese High Technology Letters*, 2017, 27(3): 203-212 (In Chinese)

[35] Zhao X, He H, Yang D, et al. Application of improved convolution neural network in image classification [J]. *Chinese High Technology Letters*, 2018, 28(11-12): 50-56 (In Chinese)

[36] Hu S, Zhao Y, Sun X. A survey of target tracking algorithm based on convolutional neural network[J]. *Chinese High Technology Letters*, 2018, 28(3): 27-33 (In Chinese)

[37] Ji X, Wu F. Head detection method based on mixed convolution neural network [J]. *Chinese High Technology Letters*, 2018, 28(4): 313-319 (In Chinese)

Li Shaojie, born in 1995. He received his B. S. from Tiangong University in 2016. He is currently pursuing his M. S. degree in the School of Microelectronics, University of Chinese Academy of Sciences. His research interests include knowledge graphs, artificial intelligence, natural language processing.