Knowledge-Based Systems 227 (2021) 107188

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Kernel multi-attention neural network for knowledge graph embedding

Dan Jiang, Ronggui Wang, Juan Yang *, Lixia Xue

School of Computer and Information, Hefei University of Technology, Hefei 230601, China

ARTICLE INFO

Article history:
Received 28 November 2020
Received in revised form 28 May 2021
Accepted 30 May 2021
Available online 4 June 2021

Keywords:
Link prediction
Entity kernel
Relation kernel
Multi-attention neural network
Knowledge graph embedding

ABSTRACT

Link prediction is the problem of predicting missing link between entities and relations for knowledge graph. In recent years, some tasks have achieved great success for link prediction, but these tasks are far from expanding entity relation vectors, and cannot predict missing links more efficiently. In this paper, we propose a novel link prediction method called kernel multi-attention neural network for knowledge graph embedding (KMAE) which is able to extend kernel separately in entity and relation attributes. The kernel function uses Gaussian kernel function to expand into more robust entity kernel and relation kernel. In addition, we constructed a novel multi-attention neural network that acts on the entity kernel and relation kernel which can capture local important characteristics. Experiments on FB15k-237 and WN18RR, show that multi-attention fully reflect excellent performance in the task of knowledge graph embedding. Our proposed KMAE achieves better results than previous state-of-the-art link prediction methods.

1. Introduction

The knowledge graph [1] utilizes its powerful semantic processing and organizational capabilities to be an indispensable boost for the advancement of Internet technology intelligence. The application of knowledge graph is extremely wide, including machine question answering, recommendation system [2,3], e-commerce, natural language processing, mobile edge computing [4,5], etc. The unit of knowledge graph is usually represented [6] by (entity 1, relation, entity 2) and (entity, attribute, attribute value) triples. However, there are missing links in some knowledge graphs, which requires prediction of corresponding links. The current knowledge graph has huge information and complex relations. Traditional link prediction methods can no longer meet the requirements of the existing knowledge graph completion tasks. In recent years, researchers have proposed some effective link prediction methods [7,8] for knowledge graph embedding.

For knowledge graph embedding, the classic TransE [9] model studies the embeddings of entities and relations of multi-relational data in low-dimensional vector space. TransE expresses entities and relations through distributed dense vectors. A relation is understood as an edge connecting the head entity and tail entity, such as (h, r, t). h, r, t represent head, relation and tail respectively. The relation between the three vectors is $h + r \approx t$. Although the TransE model is fast in training and easy to implement, it cannot solve the problems of many-to-one and one-to-many relations. In order to solve this problem, Zhen Wang propose TransH [10], which makes a good balance between capacity and efficiency. Guoliang Ji et al. believes that each relation in TransE and TransH corresponds to only one semantic representation. But in actual situations, the relation r may represent different meanings. Guoliang Ji et al. propose that TransD [11] uses two vectors to represent two entities (head entity and tail entity). One of the vectors represents the entity relation, and the other is used to construct the dynamic mapping matrix. The knowledge graph embedding models based on the Trans series also include TransR [12] and TransG [13].

The classic knowledge graph embedding model is more than the above, such as DistMult [14] and ComplEx [15]. ComplEx was inspired by DistMult to solve the problem of not being able to deal with vector asymmetric relations well. Each entity and relation in ComplEx is represented by a complex vector. Dai Quoc Nguyen et al. [16] propose the ConvKB using convolutional neural networks (CNN). ConvKB has made good progress by using CNN to capture global relations and transition features. The feature extraction capabilities of shallow models are often insufficient. Therefore, Tim Dettmers et al. [17] propose an efficient and fast-computing 2D convolutional neural network (ConvE) for knowledge graph embedding. The convolutional neural network model can solve the problem of non-linear data partition. Therefore, we are inspired by ConvE to use convolutional neural networks to predict missing links. However, this cannot expand

* Corresponding author.
E-mail address: yangjuan_hfut@126.com (J. Yang).

https://doi.org/10.1016/j.knosys.2021.107188
0950-7051/

the data feature dimension and obtain the local difference of feature attributes.

In order to solve the above problems, we study the kernel graph attention network (KGAT) [18] model. KGAT proposes a neural matching kernel for fact verification. We propose to use Gaussian kernel function to create high-dimensional spatial discernibility of entity relation attributes to form an extended kernel vector. The purpose of knowledge graph embedding is to map entities and relations to low-dimensional vector space, which is a dimension reduction technique. This technique can easily obtain the vector representation of the triples, so that the link prediction can proceed smoothly. However, it is not easy to perform link prediction in a low-dimensional dense vector space. Therefore, we embed the kernel function into the triples of entity or relation vectors. Kernel embedding improves the calculation of the dimension of entity and relation vectors, and has great effectiveness for subsequent link prediction. As far as we know, this is the first time that kernel has been expanded in entity and relation attributes of knowledge graph. Sanghyun Woo et al. propose CBAM [19] attention mechanism, which can significantly improve prediction performance of neural networks. Therefore, we propose to establish a multi-attention mechanism to improve the interpretability of the neural network and capture part of the important information in the data training process. Multi-attention is mainly composed of channel attention and spatial attention in series. Spatial attention [20] focuses on transforming the original knowledge graph relation into another space to preserve important information. The importance of channel attention [21] is that the values of different channels are multiplied by different weights, which makes key channels get more attention and is of great help to the subsequent prediction of tail entities. Combining the kernel embedding and multi-attention mechanism, we innovatively propose kernel multi-attention neural network(KMAE) model for knowledge graph embedding. The major contribution of this paper are listed as follows:

(1) We innovatively expanded entity embedding to entity kernel and relation embedding to relation kernel. In the experiment, we proved the effectiveness of expanding kernel.

(2) We propose a novel multi-attention mechanism, which mainly relies on the combination of multiple different forms of channel and spatial attention.

(3) We innovatively adopt two different attention mechanisms when using the spatial attention model. And we combine the expanded kernel to form a kernel multi-attention knowledge graph embedding model.

The rest of this paper is organized as follows. Section 2 introduces the related work. In Section 3, we propose a novel knowledge graph embedding method named KMAE. In Section 4, we conduct extensive experiments to compare KMAE with other state-of-the-art knowledge graph embedding methods. We make a conclusion and point out our future work in Section 5.

2. Related work

In recent years, link prediction methods have improved significantly. In order to solve the problem that most link prediction methods do not consider the subtle differences between relation paths, Yantao Jia et al. [22] propose a knowledge graph embedding method that minimizes path-specific edge-based loss functions. In 2019, Niannian Guan et al. [23] propose a concept embedding knowledge graph (KEC) model based on the common sense concept information of entities in the concept map, which embeds the concepts of entities and entities into the semantic space together. In 2020, Peru Bhardwaj et al. [24] study its security by designing adversarial attacks on knowledge graph embedding models. However, these knowledge graph embed models do not perform better in deeper extraction features.

In order to better extract features, the neural network model is a good object. Some neural network link prediction models have also made great progress. For example, in order to extract deep features and strong dependencies between nodes, Tim Dettmers et al. [17] propose ConvE, which is a 2D convolutional neural network model. And Keyulu Xu et al. [25] propose jumping knowledge (JK) networks. Since there is no structural enforcement in the embedding space of ConvE, Chao Shang et al. [26] propose an end-to-end structure-aware convolutional network (SACN) model. Md. Rezaul Karim et al. [27] used various embedding methods to embed nodes into the graph, and propose a fusion of convolution network and classic machine learning prediction model, which can better distinguish drug interactions. The three-level learning lacks ability to capture the long-term dependencies between entities. Therefore, Lingbing Guo et al. [28] propose recurrent jump networks (RSNs), which uses a jump mechanism to bridge the gap between entities. The disappearance of gradients on discrete data is an inherent problem of traditional generative adversarial networks (GANs). Yuanfei Dai et al. [7] propose a knowledge graph representation learning model based on adversarial networks, which introduces Wasserstein distance to solve the divergence problem of traditional models. Bonaventure C et al. [29] propose a unique hybrid model, which uses knowledge graph embedding and convolutional neural network for representation learning. Relation-aware inception network (ReInceptionE) [30] adopts local–global structural information for knowledge graph embedding. ReInceptionE overcomes the problem that there is no structural information in the embedded space of ConvE and its performance is still limited by the number of interactions. Feihu Che et al. [31] propose parameters as relation embedding (ParamE), which uses neural network parameters as relation embedding to make ParamE more expressive and translatable than ConvE.

Although neural network models have strong nonlinear fitting capabilities, they still cannot expand the spatial relation between features. The kernel function can capture the spatial feature scale of similarity between samples. Alberto Bietti et al. [32] utilize the kernel functions to regularize deep neural networks. Xiang Li et al. [33] propose deep network termed selective kernel networks (SKNets). Zhenghao Liu et al. [18] propose kernel graph attention network (KGAT), which uses kernel-based attention for more fine-grained fact verification. The kernel neural network is applied to knowledge graph embedding, which makes the entity relation mapping more comprehensive. However, this creates a complex data structure. The attention mechanism allows selective attention to more noticeable features during model training.

In order to construct attention mechanism in knowledge graph embedding, Wei Qian et al. [34] propose an effective model that can learn the real attention that changes between relations. The triples are processed separately in the knowledge graph embedding, but it cannot cover complex and hidden information. Therefore, Deepak Nathani et al. [35] propose a new attention-based feature embedding method, which can capture entity and relation features in any given entity neighborhood. The existing entity alignment methods usually ignore the knowledge of neighborhood subgraphs between entities. Qiannan Zhu et al. [36] propose a neighborhood-aware attention representation method for multilingual knowledge graphs. This method designs an attention mechanism that aggregates and learns neighbor layer representations through weighted combination. Seungwhan Moon et al. [37] propose a new attention model that learns the symbolic transformation of dialog context. The model uses the graph path decoder of attention, which predicts the introduction of natural entities into the previous dialog context. Qi Wang et al. [38] propose a deep reinforcement learning framework for multi-hop relation paths based on attention (ADRL). ADRL learns the structured perception of deep learning and the relational reasoning of reinforcement learning, which improves the efficiency, generalization ability and interpretability of traditional methods. In order to mine users' potential preferences from the high-order connectivity structure of heterogeneous knowledge graphs, Zuoxi Yang et al. [39] propose a hierarchical attention graph convolutional network containing interpretable recommended knowledge graphs. We are greatly inspired by these knowledge graph embedding models using attention mechanisms. Therefore, we combined kernel and attention neural network to study knowledge graph embedding.

In link prediction, we first adopt an embedding method to convert entity and relation attributes into triple vector representations. And we adopt the convolutional neural network method to predict the tail entity by combining the embedding matrix of the head entity vector and relation vector. The two groups of channel attention and spatial attention cascaded form are easier to capture high-quality feature vector information. This attention mechanism is called multi-attention mechanism. Therefore, we adopt the kernel function and multi-attention neural network in the knowledge graph embedding. We establish a knowledge graph embedding model based on kernel double attention neural network for preliminary experimental verification. On the basis of double attention, we propose a knowledge graph embedding model of kernel multi-attention convolutional neural network. This innovative setting has been effectively verified in a series of experiments.

3. Our proposed method

In this section, KMAE framework will be introduced in detail. We will give the definition of KMAE in Fig. 1 of this paper. A knowledge graph is usually composed of triples, which are divided into two types of entities and relations. The entities contain head entity and tail entity. The triple is formulated as (h, r, t). Among them, $h, t \in E$, E represents the entity set, $r \in R$, R represents the relation set. Next, we will first introduce extended kernel into entity and relation attributes.

3.1. Extended kernel

In this subsection, we expand the entity and relation kernel in order to make the prediction effect of neural network more significant. The importance of adding kernel functions depends on two aspects. The first aspect is that entity and relation embedding is actually a dimension reduction operation. After this operation, the data is too dense and not easy for link prediction. The kernel function is added to solve this problem ingeniously. The second aspect is to use the most appropriate embedding method for different datasets to obtain the best link prediction results. The structure of kernel utilizes the Gaussian kernel function. The Gaussian kernel function is usually called radial basis function (RBF). Its main function is to map each entity or relation sample point to infinite dimensions, making the entity or relation sample linearly separable. Eq. (1) is the definition of Gaussian kernel function.

$$K(x, x_u) = e^{-\frac{\|x-x_u\|^2}{2\sigma^2}} \tag{1}$$

The above formula is calculated from the Euclidean distance between the vectors x and x_u. Among them, σ controls the radial bandwidth. The larger the value, the greater the local influence range of the Gaussian function. x_u can control the center position of the Gaussian function graph. In the model calculation, if x_u is fixed, the change of x will have a significant impact on the result.

For the head h in the knowledge graph triplet (h, r, t), we expand the kernel, as shown below:

$$\phi(e_h) = e_h - Kernel(e_h) \tag{2}$$

$$K(e_h, kernel_u) = e^{-\frac{\|e_h-kernel_u\|^2}{2\sigma^2}} \tag{3}$$

Eq. (2) reflects the expansion of the entity kernel to the entity embedding, using $\phi(e_h)$ to represent the Gaussian function transformation of each entity node. e_h represents the entity embedding. Word embedding is the conversion of words or phrases in natural language into vector of real numbers that can be recognized by computers. Therefore, the principles of entity embedding and relation embedding in the knowledge graph are similar to word embedding. We used torch.nn.Embedding in the entity embedding and relation embedding. The main parameters in entity embedding are the number of input entities m and the dimension n of each entity word. Where n is the number of columns in each row, and each row represents a word. The number of entities in the entity attribute is multiplied by the expected dimension of each entity, and the vector formed in this way is the entity embedding. The same is true for relation embedding vectors. Eq. (3) is based on Eq. (1). The values of $kernel_u$ and σ are calculated according to the size of the kernel value k.

The relation embedding is extended to the relation kernel embedding with the RBF function, as shown below:

$$\phi(e_r) = e_r - Kernel(e_r) \tag{4}$$

$$K(e_r, kernel_u) = e^{-\frac{\|e_r-kernel_u\|^2}{2\delta^2}} \tag{5}$$

Kernel uses Gaussian kernel to extract information, and performs the kernel conversion of entity embedding e_h and relation embedding e_r [18]. The combination of entity kernel embedding $\phi(e_h)$ and relation kernel embedding $\phi(e_r)$ forms entity relation kernel embedding $\phi(e_{h\_r})$. The three kernel embedding are respectively combined with the attention neural network module in the experiments for link prediction.

3.2. Multi-attention module

In this section, we will introduce the construction of a multi-attention neural network. The multi-attention mechanism is divided into two parts. In the first part, double attention uses a combination of channel attention and spatial attention mechanisms. In the second part of the attention mechanism, the spatial attention convolution layer adds normalization processing or uses a nonlinear activation function. Fig. 2 is internal structure of the multi-attention neural network.

After the entity embedding in the previous section is transformed into entity kernel embedding, entity kernel embedding and relation embedding are connected as $[\phi(\overline{e_h}); \overline{e_r}]$. The entity embedding and relation kernel embedding are connected as $[\overline{e_h}; \phi(\overline{e_r})]$. $\bar{\cdot}$ denotes 2D reshaping of real vector. The initial input of the multi-attention neural network is e_{en_rel}. e_{en_rel} is the connection between the entity and relation vector. $e_{en\_rel} = [\phi(\overline{e_h}); \overline{e_r}]$ or $[\overline{e_h}; \phi(\overline{e_r})]$.

e_{en_rel} is normalized to make the training process more stable. The normalization process is shown in the following formula:

$$e_{en\_rel} = \frac{e_{en\_rel} - E(e_{en\_rel})}{\sqrt{Var[e_{en\_rel}] + \varepsilon}} * \gamma + \beta \tag{6}$$

Among them, γ and β are obtained through independent learning when training the network. The normalized e_{en_rel} accelerates the convergence speed of model during neural network training. In order to prevent over-fitting, we use dropout to regularize entity or relation kernel embedding. The entity or

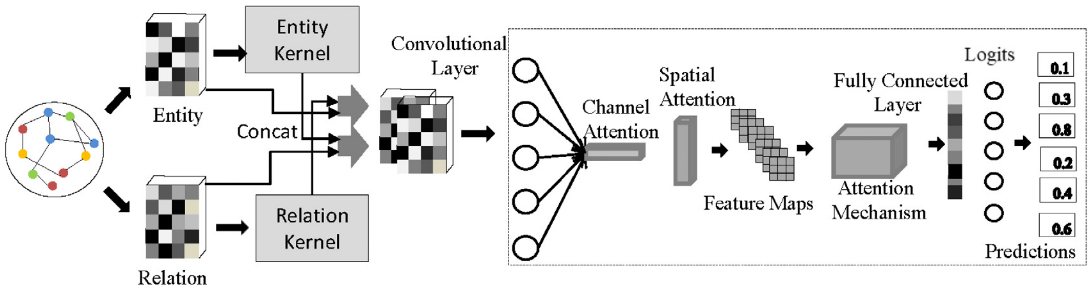

Fig. 1. The framework of kernel multi-attention for knowledge graph embedding. The entity embedding and relation embedding in the figure are respectively expanded into entity kernel and relation kernel. Entity kernel or relation kernel combines relation embedding or entity embedding input into multi-attention convolutional neural network for training.

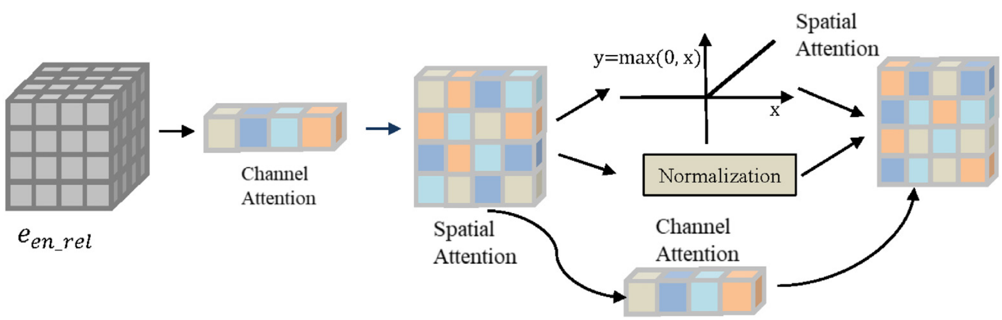

Fig. 2. Multi-attention neural network module. The multi-attention neural network module is based on the double attention neural network. The multi-attention mechanism repeatedly uses the channel attention and normalized spatial attention of the double attention module.

relation kernel embedding is processed to get the final e_{en_rel} to enter the convolutional layer for training. After the convolutional layer, e_{en_rel} is trained by the channel attention network, and the spatial attention network is used for training in series. We call such attention as the double attention module (KMAE-double). In addition, KMAE-double is effectively verified in the experiment. The main steps of KMAE-double are as follows:

$$H_{ca} = ca(e_{en\_rel}) * e_{en\_rel} \tag{7}$$

$$dou(e_{en\_rel}) = sa(H_{ca}) * H_{ca} \tag{8}$$

H_{ca} indicates that e_{en_rel} is trained in the channel attention mechanism, and $dou(e_{en\_rel})$ indicates the result obtained after the channel and spatial attention. After channel attention, we use the spatial attention mechanism in series. The spatial attention is transferred from one space to another, and the key information of the original sample is retained, thereby capturing the area that needs attention in the sample information. At the same time, it can have the functions of rotation, scaling and transformation, so that important local information can be extracted through transformation.

$dou(e_{en\_rel})$ is trained by the activation function Relu, $y = max(0, x)$, which makes dou_{en_rel} non-linear. In terms of the amount of calculation, the training process is also optimized. $dou(e_{en\_rel})$ is called $F_{e_{en\_rel}}$ after feature mapping. Then the attention mechanism performs two concatenation processing on $F_{e_{en\_rel}}$. The two most critical steps are as follows: $M_{ca} = ca(F_{e_{en\_rel}}) * F_{e_{en\_rel}}$ and $Mul_{e_{en\_rel}} = normsa(M_{ca}) * M_{ca}$. $normsa()$ is normalized spatial attention. Normalized spatial attention is obtained on the basis of spatial attention. The normalized spatial attention is more linearly separable than the spatial attention in the double attention module, which can better distinguish the subtle gaps between the values. Normalized spatial attention adds activation function Relu and normalization operation on the basis of spatial attention. Eq. (9) is a summary of these two steps.

$$Mul_{e_{en\_rel}} = Mul(F_{e_{en\_rel}}) * F_{e_{en\_rel}} \tag{9}$$

Table 1
The score functions $f_r(e_h, e_t)$ of several link prediction models, Where g denotes a nonlinear function. $*$ denotes convolution operation, and ω denotes the convolution kernel. $\|$ denotes either, which means one of the two. $\bar{\cdot}$ denotes 2D reshaping of real vector.

Models	Scoring function $f_r(e_h, e_t)$	Parameters
TransE [9]	$\|e_h + e_r - e_t\|_p$	$h, r, t \in R^n$
DistMult [14]	$\langle e_h, e_r, e_t \rangle$	$h, r, t \in R^n$
ComplEx [15]	$\langle e_h, e_r, e_t \rangle$	$h, r, t \in C^k$
ConvE [17]	$g(vec(g([\overline{e_h}; \overline{e_r}] * \omega))W)e_t$	$h, r, t \in R^n$
ConvKB [16]	$concat(g([e_h, e_r, e_t] * \omega))\beta$	$h, r, t \in R^n$
KMAE	$g(Mul([\phi(\overline{e_h}); \overline{e_r}] \parallel [\overline{e_h}; \phi(\overline{e_r})]))e_t$	$h, r, t \in R^n$

3.3. Scoring function

Attention convolutional neural networks in KMAE-double and KMAE use $dou(\cdot)$ and $Mul(\cdot)$ respectively. This scoring function for KMAE-double and KMAE is defined as:

$$g(dou([\phi(\overline{e_h}); \overline{e_r}] \parallel [\overline{e_h}; \phi(\overline{e_r})]))e_t \tag{10}$$

$$g(Mul([\phi(\overline{e_h}); \overline{e_r}] \parallel [\overline{e_h}; \phi(\overline{e_r})]))e_t \tag{11}$$

Table 1 is our summary of the scoring function of several state-of-the-art knowledge graph embedding model. It is known from Section 3.1 that $\phi(e_h)$ represents kernel embedding transformation of the entity embedding. $[\phi(\overline{e_h}); \overline{e_r}]$ denotes that $\phi(\overline{e_h})$ and $\overline{e_r}$ are connected. $Mul(\cdot)$ represents the attention network, which not only represents the multi-attention mechanism, but also contains operations such as convolution, full connection and other operation in the neural network.

In summary, KMAE adds the kernel function [18], which makes the entity or relation vector turn into the entity kernel or relation kernel. The innovative fusion of kernel function and entity relation vector increases the scalability of the knowledge graph embedding training process, and the effectiveness of adding kernel embedding is verified in subsequent experiments. Algorithm 1 embodies the specific algorithm process of KMAE. The kernel

Table 2
Summary of datasets.

Datasets	FB15k-237	WN18RR
Entities	14,541	40,943
Relations	237	11
Train edges	272,115	86,835
Valid edges	17,535	3034
Test edges	20,466	3134

multi-attention neural network has obtained good experimental results in link prediction.

Algorithm 1 Kernel Multi-attention Neural Network for Knowledge Graph Embedding (KMAE)

Input: Knowledge graph triples representative elements (h, r, t), head entity embedding e_h, relation embedding e_r, tail entity embedding e_t.
Input: Convolution kernel size = 3; niter = 1000; Gaussian kernel = 10.
1. Calculate the entity kernel $\phi(e_h)$ and relation kernel $\phi(e_r)$, using Eqs. (3) and (5).
2. Use the method of matrix connection to construct convolutional layer input vector e_{en_rel} in Section 3.2.
3. Calculate channel attention H_{ca} and spatial attention $sa(H_{ca})$ mechanism values.
4. Calculate feature mapping $F_{e_{en\_rel}}$.
5. Calculate channel attention M_{ca} and normalized spatial attention $normsa(M_{ca})$.
6. Use the sigmoid activation function to get the predicted value.
7. **repeat:** Repeated training to get the best predicted value.
8. **until:** converges.
9. **return**: *pred*.

4. Experiments

In this section, we mainly carry out four important parts of experiments. The first part is the influence of hyperparameters on the results of KMAE experiments. The second part is based on the evaluation index to show the results of KMAE-double, KMAE and the state-of-the-art link prediction method in line figure. This part of the experiment is a summary and refinement of the entire experiment. The third part is the experimental results of KMAE comparing several link prediction methods including KMAE-double. The fourth part is the experimental results of KMAE-double and KMAE under different kernel. After four series of experiments, we have concluded that KMAE is superior to KMAE-double and several comparison methods in several performances.

4.1. Datasets

We adopt two representative public datasets for our evaluation. A brief summary about FB15k-237 and WN18RR datasets are provided in Table 2.

FB15k-237 [40] is a subset of FB15k [9]. On the basis of FB15k, the inverse relation is removed to form FB15k-237. Dataset FB15k is a subset of Freebase. Therefore, FB15k-237 is a dataset obtained after multiple screenings, which is more in line with the needs of link prediction.

WN18RR [17] is a subset of WN18 [9]. The source of WN18RR is similar to that of FB15k-237. It comes from WN18, and WN18 is a subset of WordNet. WN18RR is the same as WN18 in entities, with 40 943 entities. In addition, the number of other attribute values is a little less than WN18.

4.2. Comparing methods

In this section, we employ five methods as baselines for comparison. These five methods are TransE [9], DistMult [14], ComplEx [15], R-GCN [41], ConvE [17]. Our KMAE-double and KMAE are not only compared with the classic link prediction methods TransE, DistMult, ComplEx, but also compare with the state-of-the-art R-GCN and ConvE, and have obtained good results.

4.3. Evaluation metrics

The triples (h, r, t) in link prediction usually use the head entity and relation to find the tail entity $(\mathrm{h}, \mathrm{r}, t')$, or through $(h', \mathrm{r}, \mathrm{t})$ to get h. In the experiment, Hits at N (H@N) reflects the top scores, which are represented by Hits@1, Hits@3 and Hits@10. The indicators mean reciprocal rank (MRR) and mean rank (MR) evaluation link prediction model have high reference value. Among them, MRR is widely used in questions that allow multiple results to be returned. The system scores each returned result, and then sorts them according to the score, and the value with the highest score is returned first.

4.4. Experimental settings

We choose hyperparameters for our models KMAE-double and KMAE. The ranges of the hyperparameters via the grid search are set as follows: dropout rate {0.0, 0.1, 0.2, 0.3, 0.4}, learning rate {0.001, 0.002, 0.003, 0.004, 0.005}, batch size {32, 64, 128}, epoch {100, 200, 300, 500, 1000, 1500}, Gaussian kernel {10}, kernel size $\{3 * 3,\ 1 * 1\}$. FB15k-237 and WN18RR are divided into training set, validation set and test set. The ratios of WN18RR and FB15k-237 training, validation and test sets are approximately 0.934:0.033:0.033 and 0.877:0.057:0.066, respectively. Our model is trained on PyTorch. One iteration of FB15k-237 and WN18RR takes about 90 s and 40 s, and the training time will change with different parameters.

4.5. Effects of batch size and learning rate

In this section, we will show the performance of datasets FB15k-237 and WN18RR under the influence of hyperparameters. Through Tables 3 and 4, we can clearly see the changes of the datasets under various parameters. The bold value is the maximum value of each column.

Table 3 shows the results of FB15k-237 under different parameters. When the batch size remains unchanged, the value of learning rate = 0.003 is mostly smaller than the value of learning rate = 0.001. When the batch size is 32, 64 and 128, the performance effect when batch size = 128 is obviously not as good as the previous two. On the evaluation index MR, batch size = 64 has the smallest value, which represents the best performance. Based on the comparison of experimental values, the FB15k-237 dataset has the best performance value on the batch size of 32 or 64 and learning rate = 0.001.

Table 4 shows the performance of WN18RR under different parameters. Comparing the learning rate 0.001 to 0.005, the performance of KMAE with learning rate = 0.001 and 0.003 is significantly better than 0.002, 0.004 and 0.005. The value of test batch size = 128 mostly exceeds test batch size = 64. In this regard, we specifically studied the impact of increasing learning rate on the experimental results, and finally found that the results are best when batch size = 128, test batch size = 128 and learning rate = 0.003. In general, the effects of different learning rates and batch sizes on the two datasets are quite different. Therefore, we conducted multiple rounds of experiments to capture the most suitable parameters for each dataset.

Table 3
On FB15k-237, the effect of batch size and learning rate.

Setting		FB15k-237				
Batch size	Learn rating	Hits@10	Hits@3	Hits@1	MRR	MR
32(*testbatchsize* = 128)	0.001	0.500	**0.358**	**0.240**	**0.326**	241
	0.003	0.457	0.327	0.220	0.300	270
32(*testbatchsize* = 64)	0.001	0.498	0.356	0.239	0.324	244
64(*testbatchsize* = 128)	0.001	**0.502**	0.355	0.235	0.322	**235**
	0.003	0.486	0.344	0.228	0.313	246
128(*testbatchsize* = 128)	0.001	0.494	0.348	0.23	0.317	248
	0.003	0.487	0.344	0.228	0.313	254

Table 4
On WN18RR, the effect of batch size and learning rate.

Setting		WN18RR				
Batch size	Learn rating	Hits@10	Hits@3	Hits@1	MRR	MR
64(*testbatchsize* = 128)	0.001	**0.524**	0.461	0.402	0.441	4817
	0.002	0.518	0.452	0.398	0.435	4550
	0.003	0.522	0.460	0.407	0.443	5042
128(*testbatchsize* = 128)	0.001	0.514	0.448	0.403	0.435	4509
	0.003	0.519	**0.465**	**0.413**	**0.448**	**4441**
	0.004	0.504	0.438	0.390	0.424	5049
	0.005	0.502	0.434	0.389	0.423	4849
128(*testbatchsize* = 64)	0.001	0.514	0.455	0.408	0.441	4883
	0.003	0.505	0.435	0.383	0.418	4747

4.6. Analysis of link prediction

Figs. 3 and 4 show the specific experimental results of the datasets FB15k-237 and WN18RR, respectively. The link prediction model displays the results on the evaluation indicators under different iteration times. Figs. 3 and 4(a)–(e) respectively represent the results on the evaluation indicators Hits@1, Hits@3, Hits@10, MRR and MR. In Figs. 3 and 4, KMAE-double and KMAE are compared with each other and also compared with other state-of-the-art link prediction models.

In Fig. 3, the number of iterations is the top 180 rounds. Under the evaluation index Hits@1, KMAE is completely higher than the model ComplEx and DistMult. The convergence speed of KMAE is slightly inferior to ConvE, but after 60 rounds, the KAME value is significantly higher than ConvE. On the evaluation indicators Hits@3, Hits@10 and MRR, ComplEx, DistMult and ConvE are similar to those on Hits@1. For KMAE-double, Fig. 3(a) and (c) have a coincidence trend in the later iterations. Fig. 3(b) and (d) KMAE is higher than KMAE-double. Overall, KMAE performance is better than KMAE-double. In the evaluation index MR, the convergence speed and value of KMAE are significantly better than other link prediction methods. Therefore, KMAE has the best performance on the five evaluation indexes in FB15k-237.

In Fig. 4, the results of link prediction models are relatively close, and adding tags will make it difficult to observe clearly between the model results. Therefore, we use pure solid lines without any labeling symbols. The convergence speed of WN18RR is slower than that of FB15K-237, and there are more training iterations. Fig. 4 reflects the experimental results of the top 500 rounds of the iteration. In the evaluation index Hits@1, KMAE is slightly higher than ConvE, which is almost the same as KMAE-double. Fig. 4(b) and (c) are very similar in morphology and obviously different in value. The KMAE results of these two sub-images are higher than the other four comparison methods. On MRR, KMAE still performed the best, followed by KMAE-double, and DistMult performed worse than other methods. On MR, the early iteration result of ConvE is slightly higher than that of KMAE, but the later iteration results are not as good as KMAE. When approaching 500 rounds, KMAE has already demonstrated its advantage. Therefore, KMAE performed well and achieved relatively considerable experimental results on WN18RR.

In general, comparing the line graphs of KMAE and KMAE-double, the experimental effect of KMAE is better than that of KMAE-double. Therefore, the scoring function Eq. (11) proposed in Section 3.3 is more effective than Eq. (10). The effect of multi-attention mechanism is significantly higher than that of double attention mechanism.

4.7. Comparative results with the state-of-the-art methods

Table 5 shows the experimental results of KMAE, KMAE-double and five comparison methods on FB15k-237 and WN18RR. The evaluation indicators of Hits@10, Hits@3, Hits@1, MRR and MR are used in the table. The experimental results of KMAE and KMAE-double are shown at the bottom of the table.

First, we compare KMAE-double with five comparison methods. In FB15k-237 data, KMAE-double is better than TransE in every evaluation index. Comparing KMAE-double with DistMult, ComplEx and R-GCN, KMAE-double is significantly better than the above three models on the five indicators. Compared with ConvE, KMAE-double is better than ConvE in MR and other indicators are slightly inferior to ConvE model. On WN18RR, KMAE-double is 92% higher than TransE model in MRR. In Hits@1 and MRR indicators, KMAE-double is higher than the five comparison methods, and it is 4.9% higher than DistMult on the Hits@1 indicator. In Hits@10, KMAE-double is higher than TransE, DistMult and ComplEx. Therefore, KMAE-double has good superiority in link prediction.

Then, we compare KMAE with KMAE-double and five comparison methods. On FB15k-237, KMAE is better than TransE, DistMult, ComplEx and R-GCN. Compared with ConvE, KMAE increased by 0.2% in Hits@10, increased by 1.3% in Hits@1, and increased by 3.7% in MR. On WN18RR, KMAE significantly improved MRR by 98% compared to TransE. Compared with DistMult, Hits@1 increased by 6.4% and MRR increased by 4.2%. KMAE's Hits@1 increased by 1.2% based on ComplEx data. Compared with ConvE, Hits@1 increased by 3.8%, and MRR increased by 4.2%. Comparing the results of KMAE-double and KMAE, all evaluation indicators of KMAE are better than KMAE-double. Therefore, our KMAE has superior performance than KMAE-double. And our KMAE-double and KMAE have a significant improvement for the other five link prediction models.

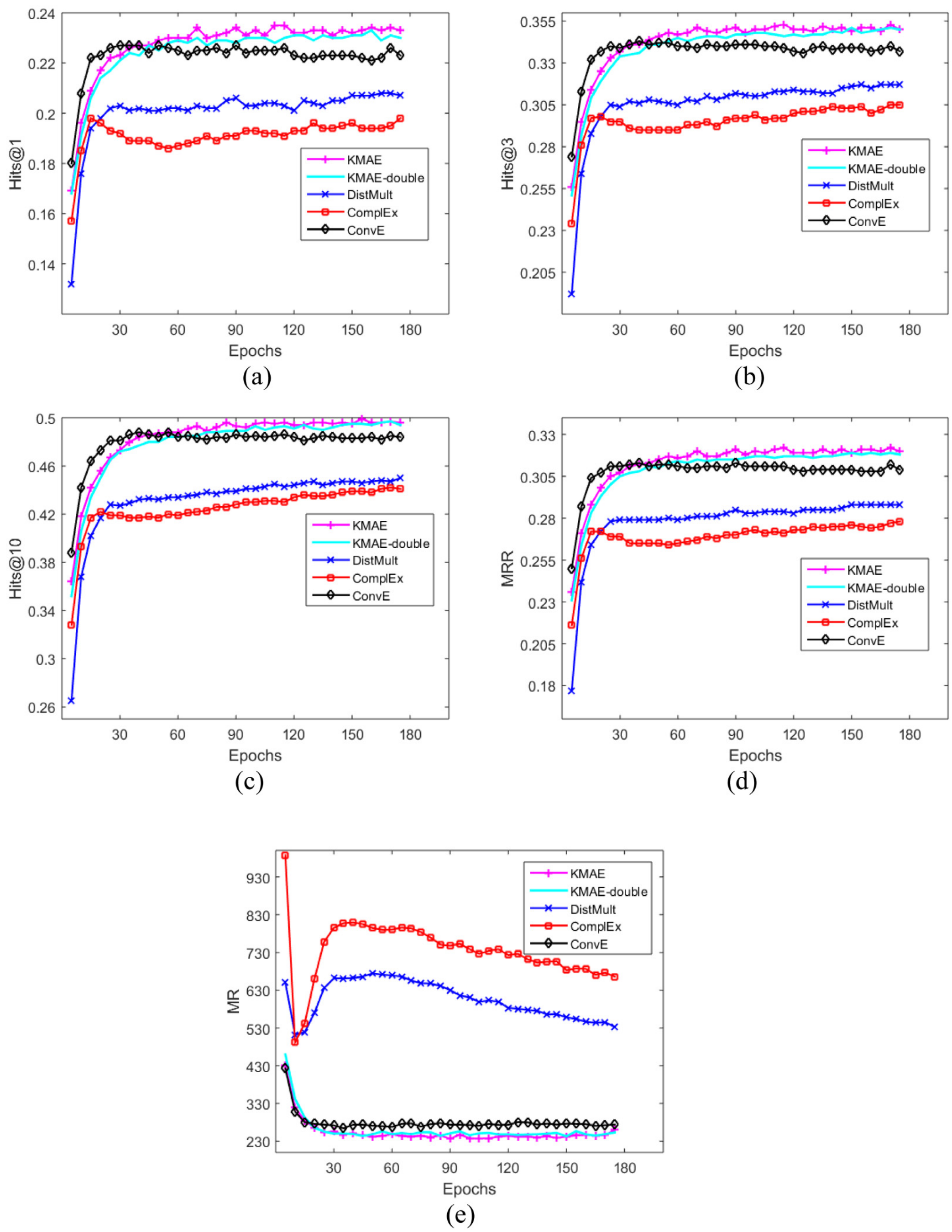

Fig. 3. The experimental results of KMAE-double and KMAE are shown and compared with the state-of-the-art knowledge graph embedding methods in FB15k-237. The kernel of Fig. 3 uses entity kernel.

Table 5
Link prediction for FB15k-237 and WN18RR. Among them, the result [⋆] are taken from [16]. Results [∗] are taken from [17].

Method	FB15k-237					WN18RR				
	Hits@10	Hits@3	Hits@1	MRR	MR	Hits@10	Hits@3	Hits@1	MRR	MR
TransE [9][⋆]	0.465	–	–	0.294	347	0.501	–	–	0.226	**3384**
DistMult [14][*]	0.419	0.263	0.155	0.241	254	0.49	0.44	0.39	0.43	5110
ConvE [17][*]	0.501	0.356	0.237	0.325	244	0.52	0.44	0.40	0.43	4187
R-GCN [41][*]	0.417	0.258	0.153	0.248	–	–	–	–	–	–
ComplEx [15][*]	0.428	0.275	0.158	0.247	339	0.51	0.46	0.41	0.44	5261
KMAE-double	0.500	0.352	0.233	0.321	242	0.511	0.446	0.409	0.434	5054
KMAE	**0.502**	**0.358**	**0.240**	**0.326**	**235**	**0.524**	**0.465**	**0.415**	**0.448**	4441

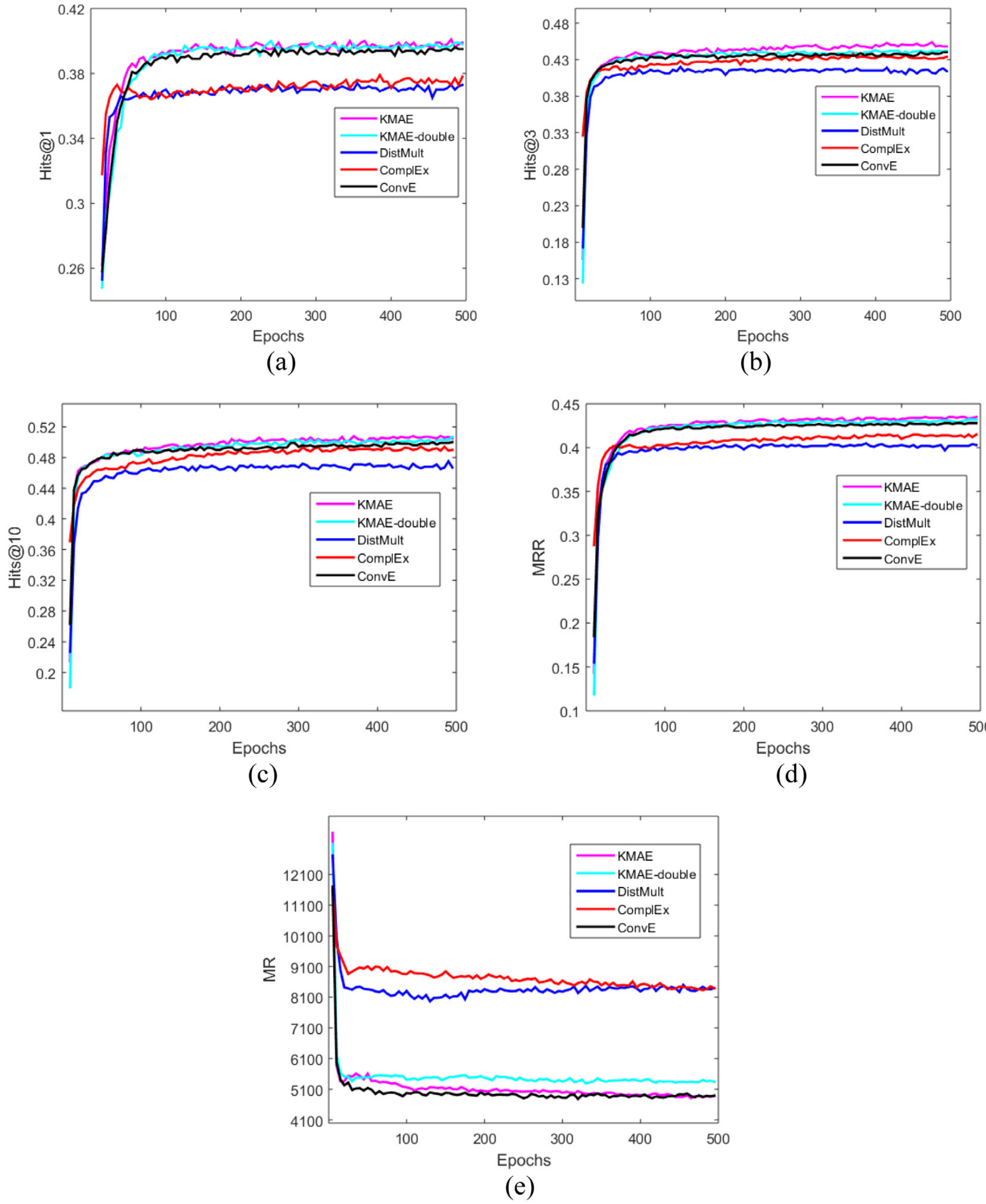

Fig. 4. The experimental results of KMAE-double and KMAE are shown and compared with the state-of-the-art knowledge graph embedding methods in WN18RR. The kernel of Fig. 4 uses relation kernel.

Table 6
Kernel embedding with double attention neural networks.

Kernel	FB15k-237					WN18RR				
	Hits@10	Hits@3	Hits@1	MRR	MR	Hits@10	Hits@3	Hits@1	MRR	MR
$Entitykernel_{doubleattention}$	**0.500**	**0.352**	**0.233**	**0.321**	**242**	0.486	0.444	**0.409**	0.434	5162
$Relationkernel_{doubleattention}$	0.485	0.342	0.227	0.312	277	**0.511**	**0.446**	0.401	**0.434**	**5054**
$En\_Relkernel_{doubleattention}$	0.487	0.344	0.229	0.314	253	0.481	0.436	0.401	0.427	5652

Table 7
Kernel embedding with multi-attention neural networks.

Kernel	FB15k-237					WN18RR				
	Hits@10	Hits@3	Hits@1	MRR	MR	Hits@10	Hits@3	Hits@1	MRR	MR
$Entitykernel_{multi-attention}$	**0.502**	**0.358**	**0.240**	**0.326**	**235**	0.491	0.449	**0.415**	0.439	4765
$Relationkernel_{multi-attention}$	0.482	0.339	0.224	0.309	294	**0.524**	**0.465**	0.413	**0.448**	**4441**
$En\_Relkernel_{multi-attention}$	0.497	0.350	0.232	0.318	249	0.481	0.439	0.405	0.430	4934

4.8. Influences of kernel module

The experimental results of entity kernel, relation kernel and entity relation kernel on multi-attention network and double attention network. We also used five evaluation indicators to measure the performance on the two datasets. Tables 6 and 7 show the effect of kernel embedding on entity or relation vector after the attention mechanism is added. Among them, Table 6 shows that in the context of double attention. In order to better show the effect of expanding the kernel, we reflect the respective experimental results of the three kernels in this subsection. $Entitykernel_{doubleattention}$ is the result of $[\phi(\overline{e_h}); \overline{e_r}]$. $Relationkernel_{doubleattention}$ is the result of $[\overline{e_h}; \phi(\overline{e_r})]$. $En\_Relkernel_{doubleattention}$ is the result of merging the above entities and relation kernels $[\phi(\overline{e_h}); \phi(\overline{e_r})]$.

Table 6 is the result obtained under the double attention neural network. On FB15k-237, the Hits@1, Hits@3, Hits@10, MRR and MR of the entity kernel are 2.6%, 2.9%, 3.1%, 2.9% and 12.6% higher than that of the relation kernel, respectively. The difference in experimental results between the entity kernel and entity relation kernel is smaller than that between the entity kernel and the relation kernel. Therefore, the experimental results of the entity kernel are higher than the relation kernel and entity relation kernel in FB15k-237. On WN18RR, the experimental values of entity kernel and relation kernel are opposite to those on FB15k-237. The experimental results of relation kernel are higher than entity kernel in most evaluation indexes on WN18RR.

Table 7 shows the performance of kernel embedding under multi-attention neural network. The evaluation index settings in Table 7 are the same as those in Table 6. In FB15k-237, the five evaluation index value entity kernels are higher than relation kernels. According to Tables 6 and 7, the performance of the relation kernel on FB15k-237 is inferior to entity kernel and entity relation kernel. On WN18RR, the relation kernel performs better than the other two kernels, and the experimental value of entity relation kernel is the worst. Therefore, the kernel performs best on the multi-attention neural network by comparing Tables 6 and 7.

5. Conclusion and future work

We propose a novel link prediction model which is kernel multi-attention neural network for knowledge graph embedding. KMAE expands the kernel to entity and relation kernel, which makes the work of entity and relation attributes more efficient in link prediction. KMAE shows that using different kernel embedding for corresponding datasets can achieve the best link prediction effect. Our studies illustrate the kernel contribute to different attention neural network. Moreover, we propose a novel multi-attention neural network that enables KMAE to achieve outstanding performance in experimental results. We have adopted a series of comparative experiments to compare the state-of-the-art link prediction methods to demonstrate the superiority of KMAE.

Although the KMAE method shows an encouraging result for knowledge graph embedding, some issues still remain for us to study in the future: (1) we will further investigate how to extend knowledge graph embedding to question answering system or recommendation system. (2) we will further improve the stability of our KMAE model to obtain good performance on more datasets.

CRediT authorship contribution statement

Dan Jiang: Conceptualization, Methodology, Software, Writing - original draft. **Ronggui Wang:** Validation, Formal analysis. **Juan Yang:** Writing - review & editing. **Lixia Xue:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This project was supported by a grant from the National Key R&D Program of China of the Ministry of Science and Technology of the People's Republic of China (No. 2020YFC1512601).

References

[1] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, Philip S. Yu, A survey on knowledge graphs: Representation, acquisition and applications, 2020, CoRR.

[2] Yang Xiao, Qingqi Pei, Lina Yao, Xianzhi Wang, Recrisk: An enhanced recommendation model with multi-facet risk control, Expert Syst. Appl. (2020) 113561.

[3] Yang Xiao, Qingqi Pei, Tingting Xiao, Lina Yao, Huan Liu, Mutualrcc: Joint friend and item recommendations with mutualistic attentional graph neural networks, J. Netw. Comput. Appl. 177 (2020) 102954.

[4] Jie Feng, F. Richard Yu, Qingqi Pei, Xiaoli Chu, Jianbo Du, Li Zhu, Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforce-ment learning approach, IEEE Internet Things J. 7 (7) (2020) 6214–6228.

[5] Lei Liu, Jie Feng, Qingqi Pei, Chen Chen, Mianxiong Dong, Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach, IEEE Internet Things J. PP (99) (2020) 1.

[6] Silvio Domingos Cardoso, Marcos Da Silveira, Cédric Pruski, Construction and exploitation of an historical knowledge graph to deal with the evolution of ontologies, Knowl.-Based Syst. 194 (2020) 105508.

[7] Yuanfei Dai, Shiping Wang, Xing Chen, Chaoyang Xu, Wen-zhong Guo, Generative adversarial networks based on wasserstein distance for knowledge graph embeddings, Knowl.-Based Syst. 190 (2020) 105165.

[8] Mrinmaya Sachan, Knowledge graph embedding compression, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, 2020, pp. 2681–2691.

[9] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 27th Annual Meeting of the Neural Information Processing Systems, NIPS, 2013, pp. 2787–2795.

[10] Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014, pp. 1112–1119.

[11] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL, 2015, pp. 687–696.

[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.

[13] Han Xiao, Minlie Huang, Xiaoyan Zhu, Transg : A generative model for knowledge graph embedding, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, 2016.

[14] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the 3rd International Conference on Learning Representations, ICLR, 2015.

[15] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, Guillaume Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on Machine Learning, ICML, Vol. 48, 2016, pp. 2071–2080.

[16] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Q. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Vol. 2, 2018, pp. 327–333.

[17] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Convolutional 2d knowledge graph embeddings, in: Proceed-ings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 1811–1818.

[18] Zhenghao Liu, Chenyan Xiong, Maosong Sun, Zhiyuan Liu, Fine-grained fact verification with kernel graph attention network, in: Proceedings of the 58th Annual Meeting of the Association for Com-putational Linguistics, ACL, 2020, pp. 7342–7351.
[19] Sanghyun Woo, Jongchan Park, Joon-Young Lee, In So Kweon, CBAM: convolutional block attention module, in: Proceedings of the 15th ECCV, Vol. 11211, 2018, pp. 3–19.
[20] Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu, Spatial transformer networks, in: Proceedings of the 28th Annual Conference on Neural Information Processing Systems, NIPS, 2015, pp. 2017–2025.
[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023.
[22] Yantao Jia, Yuanzhuo Wang, Xiaolong Jin, Xueqi Cheng, Path-specific knowledge graph embedding, Knowl.-Based Syst. 151 (2018) 37–44.
[23] Niannian Guan, Dandan Song, Lejian Liao, Knowledge graph embedding with concepts, Knowl.-Based Syst. 164 (2019) 38–44.
[24] Peru Bhardwaj, Towards adversarially robust knowledge graph embeddings, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, Vol. 34 (10), 2020, pp. 13712–13713.
[25] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Kenichi Kawarabayashi, Stefanie Jegelka, Representation learning on graphs with jumping knowledge networks, in: Proceedings of the 35th International Conference on Machine Learning, ICML, Vol. 80, 2018, pp. 5449–5458.
[26] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, Bowen Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in: Proceedings of the 33th AAAI Con-ference on Artificial Intelligence, 2019, pp. 3060–3067.
[27] Md. Rezaul Karim, Michael Cochez, Joao Bosco Jares, Mamtaz Ud-din, Oya Deniz Beyan, Stefan Decker, Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network, in: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB, 2019, pp. 113–123.
[28] Lingbing Guo, Zequn Sun, Wei Hu, Learning to exploit long-term relational dependencies in knowledge graphs, in: Proceedings of the 36th International Conference on Machine Learning, ICML, Vol. 97, 2019, pp. 2505–2514.
[29] Bonaventure C. Molokwu, Ziad Kobti, Social network analysis using RLVECN: representation learning via knowledge-graph embed-dings and convolutional neural-network, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI, 2020, pp. 5198–5199.
[30] Zhiwen Xie, Guangyou Zhou, Jin Liu, Jimmy Xiangji Huang, Reinceptione: Relation-aware inception network with joint local–global structural information for knowledge graph embedding, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, 2020, pp. 5929–5939.
[31] Feihu Che, Dawei Zhang, Jianhua Tao, Mingyue Niu, Bocheng Zhao, Parame: Regarding neural network parameters as relation embeddings for knowledge graph completion, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 2774–2781.
[32] Alberto Bietti, Grégoire Mialon, Dexiong Chen, Julien Mairal, A kernel perspective for regularizing deep neural networks, in: Proceedings of the 36th International Conference on Machine Learning, ICML, Vol. 97, 2019, pp. 664–674.
[33] Xiang Li, Wenhai Wang, Xiaolin Hu, Jian Yang, Selective kernel networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 510–519.
[34] Wei Qian, Cong Fu, Yu Zhu, Deng Cai, Xiaofei He, Translating embeddings for knowledge graph completion with relation attention mechanism, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI, 2018, pp. 4286–4292.
[35] Deepak Nathani, Jatin Chauhan, Charu Sharma, Manohar Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL, Volume 1: Long Papers, 2019, pp. 4710–4723.
[36] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, Li Guo, Neighborhood-aware attentional representation for multilingual knowledge graphs, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 1943–1949.
[37] Seungwhan Moon, Pararth Shah, Anuj Kumar, Rajen Subba, Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs, in: Proceedings of the 57th Con-ference of the Association for Computational Linguistics, ACL, Volume 1: Long Papers, 2019, pp. 845–854.
[38] Qi Wang, Yongsheng Hao, Jie Cao, ADRL: an attention-based deep reinforcement learning framework for knowledge graph reasoning, Knowl.-Based Syst. 197 (2020) 105910.
[39] Zuoxi Yang, Shoubin Dong, Hagerec: Hierarchical attention graph convolutional network incorporating knowledge graph for explainable recommendation, Knowl.-Based Syst. 204 (2020) 106194.
[40] Kristina Toutanova, Danqi Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 57–66.
[41] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, Max Welling, Modeling relational data with graph convolutional networks, in: Proceedings of the Semantic Web 15th International Conference ESWC, Vol. 10843, 2018, pp. 593–607.