Improving Knowledge Graph Embedding Using Locally and Globally Attentive Relation Paths

Ningning Jia, Xiang Cheng(✉), and Sen Su

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications,
Beijing, People's Republic of China
chengxiang@bupt.edu.cn

Abstract. Knowledge graphs' incompleteness has motivated many researchers to propose methods to automatically infer missing facts in knowledge graphs. Knowledge graph embedding has been an active research area for knowledge graph completion, with great improvement from the early TransE to the current state-of-the-art ConvKB. ConvKB considers a knowledge graph as a set of triples, and employs a convolutional neural network to capture global relationships and transitional characteristics between entities and relations in the knowledge graph. However, it only utilizes the triple information, and ignores the rich information contained in relation paths. In fact, a path of one relation describes the relation from some aspect in a fine-grained way. Therefore, it is beneficial to take relation paths into consideration for knowledge graph embedding. In this paper, we present a novel convolutional neural network-based embedding model PConvKB, which improves knowledge graph embedding by incorporating relation paths locally and globally. Specifically, we introduce attention mechanism to measure the local importance of relation paths. Moreover, we propose a simple yet effective measure DIPF to compute the global importance of relation paths. Experimental results show that our model achieves substantial improvements against state-of-the-art methods.

Keywords: Knowledge graph embedding · Link prediction · Triple classification · Convolutional neural network · Attention mechanism

1 Introduction

Large-scale knowledge graphs such as Freebase [3], DBpedia [1], and Wikidata [38] store real-world facts in the form of triples (*head, relation, tail*), abbreviated as (h, r, t), where *head* and *tail* are entities and *relation* represents the relationship between *head* and *tail*. They are important resources for many intelligence applications like question answering and web search. Although current knowledge graphs consist of billions of triples, they are still far from complete and missing crucial

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 17–32, 2020.
https://doi.org/10.1007/978-3-030-45439-5_2

facts, e.g., 75% of the person entities in Freebase have no known nationality [8], which hampers their usefulness in the aforementioned applications.

Various methods are proposed to address this problem, and the knowledge graph embedding methods have attracted increasing attention in recent years. The main idea of knowledge graph embedding is to embed entities and relations of a knowledge graph into a continuous vector space and predict missing facts by manipulating the entity and relation embeddings involved. Among knowledge graph embedding methods, the translation-based models are simple and efficient, also perform well. For example, given a triple (h, r, t), the most well-known translation-based model TransE [5] models the relation r as a translation vector $\mathbf{r}$ connecting the embeddings $\mathbf{h}$ and $\mathbf{t}$ of the two entities, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. It performs well on simple relations, i.e., 1-to-1 relations. but poorly on complicated relations, i.e., 1-to-N, N-to-1 and N-to-N relations. To address this issue, TransH [41], TransR [20] and TransD [14] are proposed. Unfortunately, these models are less simplicity and efficiency than TransE. Nickel et al. [26] present HolE, which uses circular correlation to combine the expressive power of the tensor product with the simplicity and efficiency of TransE.

Recently, several convolutional neural network (CNN)-based models [7,22,23] have been proposed to learn the embeddings of entities and relations in knowledge graphs, in which [22] reserves the transitional characteristic in translation-based models and is comparably simple and efficient, achieves state-of-the-art performance. However, it only focuses on knowledge triples, ignoring the rich knowledge contained in relation paths. In fact, a path of one entity pair describes the relation connecting the entity pair from some aspect in a fine-grained way, and the importance of each path is different. For example, in Fig. 1, the two paths *place of birth – country* and *friend – nationality* of entity pair *(Tom Cruise, America)* describes the relation *nationality* from the location and social way, respectively. Since the path *place of birth – country* is more essential than *friend – nationality* to express the relation *nationality*, thus it is more important from the local view. Moreover, from the global view the path *friend – nationality* also occurs in entity pair *(Tom Cruise, England)*, which is connecting by the relation *travel*, thus it is less important than the path *place of birth – country* to express the relation *nationality*.

In this paper, we present a path-augmented CNN-based model, which incorporates relation paths for knowledge graph embedding. Specifically, we first introduce the attention mechanism to automatically measure the local importance of each path for the given entity pair, then inspired by inverse document frequency, we propose degree-guided inverse path frequency to compute the global importance of each path. Finally, we improve knowledge graph embedding by incorporating locally and globally attentive relation paths.

Our contributions in this paper are summarized as follows:

- We present a path-augmented CNN-based knowledge graph embedding model, which improves embedding model by incorporating relation paths locally and globally.

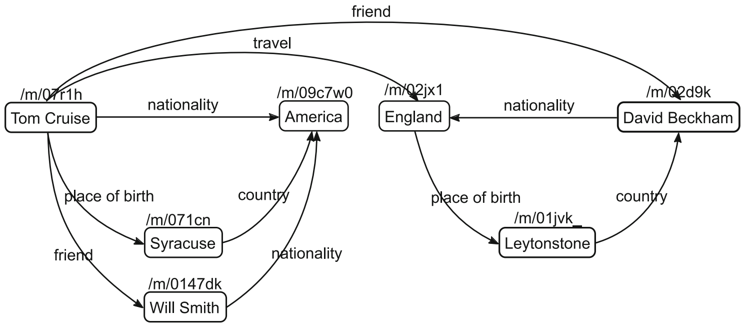

Fig. 1. An illustration that a path of one relation describes the relation from some aspect in a fine-grained way, and the importance of each path is different.

- We introduce attention mechanism to model the local importances of relation paths for knowledge graph embedding.
- We propose a simple yet effective measure, degree-guided inverse path frequency, to compute the global importances of relation paths for knowledge graph embedding.
- In addition, we apply three pooling operations to aggregate convolutional feature maps, which reduces the number of parameters greatly.
- The experimental results on four benchmark datasets show that our model achieves state-of-the-art performance.

2 Preliminaries

2.1 Problem Definition

Given a knowledge graph $\mathcal{G}$, which is a collection of valid factual triples (h, r, t), where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. $\mathcal{E}$ is the entity set and $\mathcal{R}$ is the relation set. In knowledge graph completion, embedding methods aim to define a score function f that gives an implausibility score for each triple (h, r, t) such that valid triples receive lower scores than invalid triples.

2.2 ConvKB

In this section, we briefly describe the state-of-the-art CNN-based model ConvKB, and choose it as the base of our model.

For each triple (h, r, t), ConvKB denotes the dimensionality of embeddings by k, such that each embedding triple $(\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t)$ can be viewed as a matrix $\mathbf{A} = [\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] \in \mathbb{R}^{k \times 3}$. A filter $\boldsymbol{\omega} \in \mathbb{R}^{1 \times 3}$ is repeatedly operated over every row of $\mathbf{A}$ to generate a feature map $\boldsymbol{v} = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^k$, in which $v_i = g(\boldsymbol{\omega} \cdot \mathbf{A}_{i,:} + b)$, where $\cdot$ denotes a dot product, $\mathbf{A}_{i,:}$ is the i-th row of $\mathbf{A}$, b is a bias term, and g is the non-linear activation function ReLU. In particular, if $\boldsymbol{\omega} = [1, 1, -1]$, $b = 0$, and $g(x) = |x|$ or $g(x) = x^2$, ConvKB reduces to the plain

TransE. Hence, in some point of view, ConvKB is an extension of TransE, which models triple more globally and comprehensively. The overview of ConvKB is shown in Fig. 2.

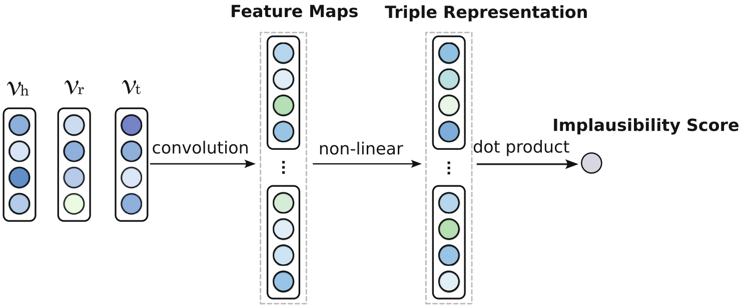

Fig. 2. The architecture of ConvKB.

Let $\boldsymbol{\Omega}$ and n denote the set of filters and the number of filters, respectively. ConvKB uses n filters to generate n feature maps. These feature maps are concatenated into a single vector, which is then calculated using the dot product with a weight vector $\mathbf{w} \in \mathbb{R}^{nk\times 1}$ to give an implausibility score for the triple (h, r, t). Formally, the score function of ConvKB is defined as follows:

$$f_{ConvKB}(h, r, t) = \text{concat}(g([\boldsymbol{v}_h, \boldsymbol{v}_r, \boldsymbol{v}_t] * \boldsymbol{\Omega})) \cdot \mathbf{w} \tag{1}$$

where $\boldsymbol{\Omega}$ and $\mathbf{w}$ are shared parameters, independent of h, r and t, $*$ denotes the convolution operator, and concat denotes the concatenation operator.

It is obvious that ConvKB only learns from triples, ignoring the rich knowledge contained in relation paths, which can lead to poor performance.

3 Our Proposed Model

3.1 PConvKB

In this section, we present our model PConvKB, which learns the embeddings by taking relation paths into consideration. Moreover, we also take into account the local and global importances of the relation paths. The architecture of our model is shown in Fig. 3.

We denote relation paths between the head entity h and the tail entity t as $P(h, t) = \{p_1, p_2, \ldots, p_N\}$, where relation path $p = (r_1, \ldots, r_m)$ is a series of interconnected relations between the entities, i.e., $h \xrightarrow{r_1} \ldots \xrightarrow{r_m} t$. Similar to ConvKB, for each triple (h, r, t), the score function of our model PConvKB is defined as follows:

$$f_{PConvKB}(h, r, t) = \sigma(\psi([\boldsymbol{v}_h, \sum_{i=1}^{N} \Phi_{G_i} \times \Phi_{L_i} \times \boldsymbol{p}_i + \boldsymbol{v}_r, \boldsymbol{v}_t] * \boldsymbol{\Omega})) \cdot \mathbf{w} \tag{2}$$

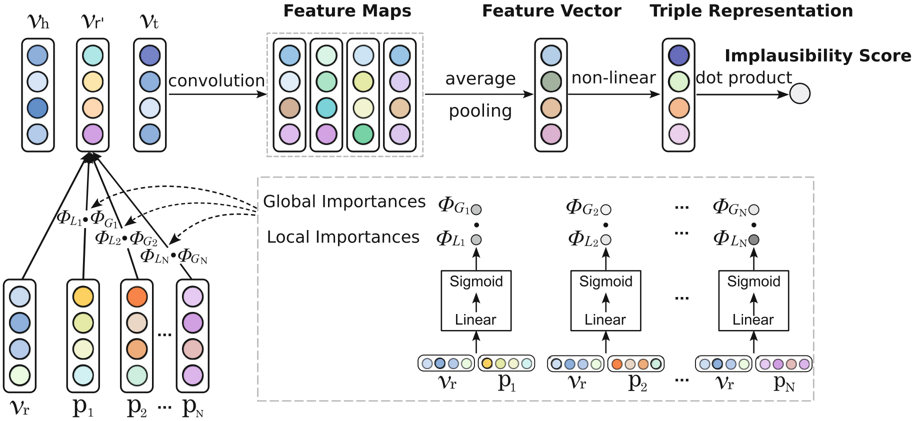

Fig. 3. The architecture of our model PConvKB.

where σ denotes the non-linear function, i.e., sigmoid, ψ denotes the average pooling operation, Φ_{G_i} denotes the global importance of the i-th path, Φ_{L_i} denotes the local importance of the i-th path, $\boldsymbol{p}_i$ is the embedding of the i-th path, which is computed as $\sum_{i=1}^{m} \boldsymbol{v}_{r_i}$, $\boldsymbol{\Omega}$ and $\mathbf{w}$ are shared parameters.

The computation of local and global importances is detailed in Sects. 3.2 and 3.3, respectively.

3.2 Measuring Local Importances of Relation Paths by Attention Mechanism

Attention mechanism [2] is designed to improve the performance of encoder-decoder model on machine translation, which assigns different weights to different data to allow the model focusing on important data. In recent years, attention mechanism has been widely used in several research topics, such as question answering [18] and image captioning [40]. In this paper, we apply attention mechanism to measure the local importances of relation paths for knowledge graph embedding. Given a triple (h, r, t) and its set of relation paths $P(h, t) = \{p_1, p_2, \ldots, p_N\}$, we compute the local importance of each path as:

$$\Phi_{L_i} = sigmoid(\boldsymbol{v}_r W_L \boldsymbol{p}_i) \tag{3}$$

where $W_L \in \mathbb{R}^{k \times k}$ is the parameter matrix. Similar to [19], we set the maximum length of each path to 3.

3.3 Measuring Global Importances of Relation Paths by Degree-Guided Inverse Path Frequency

Since the attention mechanism only focuses on the set of relation paths $P(h, t)$ of the given entity pair (h, t) that connects by the relation r. It does not consider that the path in the set of relation paths may also occur in other entity pairs that connects by other relations. Typically, the more set of relation paths a path occurs in, the less importance the path is. Therefore, inspired by inverse document frequency [10,16], which is a weighting function that has been widely used for measuring how informative each word is in a set of documents. We propose the **D**egree-guided **I**nverse **P**ath **F**requency (**DIPF**) to model the global importance of each path in the set of relation paths.

For each relation $r \in \mathcal{R}$ in the knowledge graph $\mathcal{G}$, we first find its corresponding entity pairs $(h^r, t^r)_i, i = 1, 2, \ldots, n^r$, where n^r is the number of entity pairs connecting by the relation r. i.e.,

$$(h^r, t^r)_i \in \mathcal{G} \quad and \quad (h^r, r, t^r)_i \in \mathcal{G}. \tag{4}$$

Then, we choose the entity pair $(h^r, t^r)_b$, which has the biggest node degree and is computed as:

$$\text{NodeDegree}((h^r, t^r)_b) = \max[\text{NodeDegree}((h^r, t^r)_i)], i = 1, 2, \ldots, n^r \tag{5}$$

in which,

$$\text{NodeDegree}((h^r, t^r)_i) = deg(h_i^r) + deg(t_i^r) \tag{6}$$

where NodeDegree$(\cdot)$ is the function to compute the node degree of an entity pair, max$[\cdot]$ is the maximum function, $deg(\cdot)$ is the node degree of an entity, which is computed as the number of the edges connected with the entity.

Next, we count the set of relation paths for entity pair $(h^r, t^r)_b$, and is denoted as $P((h^r, t^r)_b)$:

$$P((h^r, t^r)_b) = \{p_1^r, p_2^r, \ldots, p_{m^r}^r\} \tag{7}$$

where m^r is the number of paths of entity pair $(h^r, t^r)_b$. Similar to local importance computation, we set the maximum length of each path to 3.

Finally, the global importance of each path in the set of relation paths $P(h, t) = \{p_1, p_2, \ldots, p_N\}$ of the given triple (h, r, t) is computed as:

$$\varPhi_{G_i} = log\frac{|\mathcal{R}|}{pt_i} \tag{8}$$

where $|\mathcal{R}|$ is the cardinality of $\mathcal{R}$ (i.e., total number of relations in $\mathcal{R}$), pt_i is the number of times the path p_i occurs in the set of $\{P((h^r, t^r)_b), r \in \mathcal{R}\}$.

3.4 Aggregating Feature Maps Using Pooling Operation

As mentioned in Sect. 2.1, ConvKB uses concatenate operation to aggregate feature maps. However, previous works [30,35] demonstrate that pooling operation can better aggregate feature maps than simply concatenate operation, and reduce the number of parameters greatly. In this paper, we adopt the following three pooling operations to replace the concatenate operation, respectively:

$$\psi_{sum} = \sum_{i=1}^{n} \boldsymbol{v}_i \tag{9}$$

$$\psi_{ave} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i \tag{10}$$

$$\psi_{max} = \max([\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]) \tag{11}$$

The average pooling operation is finally chosen due to its superior performance in the experiments.

3.5 Model Training

The objective is to ensure that a triple in the golden set $\mathcal{G}$ should have a lower implausibility score than a triple in the corrupted triple set $\mathcal{G}^{'}$. Similar to [22], we adopt Adam optimizer [17] to train PConvKB, and minimize the loss function with L_2 regularization on the weight vector $\mathbf{w}$ as follows:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{G}\bigcup\mathcal{G}'} log(1 + \exp(l_{(h,r,t)} \cdot f_{PConvKB}(h,r,t))) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \tag{12}$$

in which, $l_{(h,r,t)} = \begin{cases} 1, \ for \ (h,r,t) \in \mathcal{G} \\ -1, \ for \ (h,r,t) \in \mathcal{G}^{'}. \end{cases}$

3.6 Complexity Analysis

We compare the parameter size and computational complexity of our model PConvKB with ConvKB. Let N_e denote the number of entities, N_r the number of relations, K the embedding dimension, S the number of triples for learning, P the expected number of relation paths connecting two entities, and L the expected length of relation paths. The parameter size of PConvKB is equal to the parameter size of ConvKB, i.e., $(N_e + N_r)K$. For each iteration in optimization, the computational complexity of PConvKB is $O(SKPL)$, and the computational complexity of ConvKB is $O(SK)$.

4 Experiments

For a fair comparison, we evaluate our model on two tasks: link prediction [5], and triples classification [33]. Both of them evaluate the accuracy of predicting unseen triples from different viewpoints.

4.1 Datasets

We evaluate our model on four benchmark datasets WN18 [5], FB15k [5], WN18RR [7] and FB15k-237 [36]. WN18 is extracted from WordNet [21], which contains word concepts and lexical relations between the concepts. FB15k is a subset of Freebase constructed by Bordes et al. [5]. As noted by Toutanova and Chen [36], WN18 and FB15k have problematic reversible triples causing abnormally high results. This is the reason that the refined version of WN18 and FB15k, i.e., WN18RR and FB15k-237, are widely used in state-of-the-art methods. Table 1 shows the statistics of the datasets used in our experiments.

Table 1. Statistics of the experimental datasets

Dataset	#Entity	#Relation	#Train	#Valid	#Test
WN18	40,943	18	141,442	5,000	5,000
FB15k	14,951	1,345	483,142	50,000	59,071
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,541	237	272,115	17,535	20,466

4.2 Comparison Methods

To demonstrate the effectiveness of our model, we compare PConvKB against a variety of knowledge graph embedding methods developed in recent years.

- **TransE** [5] is one of the most widely used knowledge graph embedding methods.
- **TransH** [41] associates each relation with a relation-specific hyperplane to alleviate the complex relations problem.
- **TransD** [14] not only considers the complex relations, but also the diversity of entities, by embedding entities and relations into separate entity space and relation-specific spaces.
- **HolE** [26] uses circular correlation, a novel compositional operator, to capture rich interactions of embeddings.
- **ConvE** [7] is the first CNN-based model for knowledge graph embedding.
- **ConvKB** [22] improves ConvE by taking the transitional characteristic (i.e., one of the most useful intuitions for knowledge graph completion) into consideration.
- **CapsE** [23] combines convolutional neural network with capsule network [29] for knowledge graph embedding.

4.3 Link Prediction

Link prediction task is to complete a triple (h, r, t) with h or t missing, i.e., to predict the missing h given (r, t) or the missing t given (h, r).

Evaluation Protocol. To evaluate the performance in link prediction, we follow the standard protocol used in [5]. For each test triple (h, r, t), we replace either h or t by each of other entities in $\mathcal{E}$ to create a set of corrupted triples, and calculate implausibility scores on the corrupted triples. Ranking these scores in ascending order, we can get the rank of the test triple. Notice that a corrupted triple may exist in train, validation or test set, we use the Filtered setting protocol [5] to eliminate its misleading effect, i.e., not taking any corrupted triples that appear in the knowledge graph into accounts. We employ two common evaluation metrics: mean rank (MR) and Hits@10. MR is the mean of the test triples' ranks. Hits@10 is the percentage of test triples that are ranked within top 10.

Implementation Details. Following the previous work [41], we use the common Bernoulli trick to generate the head or tail entities when sampling invalid triples. Like in ConvKB [22], we also use entity and relation embeddings produced by TransE to initialize entity and relation embeddings in PConvKB. We use the pre-trained 100-dimensional glove word embeddings [28] to train TransE model, and employ the TransE implementation provided by [25]. We select the learning rate in $\{5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}\}$, the number of filters in $\{50, 100, 200, 400\}$. We fix the batch size at 128 and set the L_2-regularizer λ at 0.001 in our objective function. We run PConvKB up to 150 epochs and monitor the Hits@10 score after every 10 training epochs to choose optimal hyper-parameters. We obtain the highest Hits@10 scores on the validation set when learning rate at $5e^{-5}$, the number of filters at 400 on WN18; and learning rate at $1e^{-5}$, the number of filters at 50 on FB15k; and the learning rate at $5e^{-6}$, the number of filters at 400 on WN18RR; and the learning rate at $1e^{-5}$, the number of filters at 200 on FB15k-237. For comparison methods, we use the codes released by [11], [7] and [22].

Table 2. Experiments results on link prediction. Hits@10 is reported in %. The best score is in **bold**, while the second best score is in underline. For comparison methods, the values in black color are the results listed in the original publication, except ConvKB uses the [23] implemented version, which has been reported significantly better performance than the original one. The values in blue color are obtained by implementations from the OpenKE repository.

Model	WN18		FB15k		WN18RR		FB15k-237	
	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
TransE	–	–	125	47.1	–	–	–	–
	733	60.3	168	60.6	5766	32.4	349	42.2
TransH	388	82.3	87	64.4	–	–	–	–
	602	87.7	137	63.1	7019	36.3	358	40.0
TransD	212	92.2	91	77.3	–	–	–	–
	617	88.5	151	62.9	7050	36.8	418	40.2
HolE	–	94.9	–	73.9	–	–	–	–
	589	79.6	76	76.2	5992	38.5	369	43.3
ConvE	504	95.5	64	87.3	5277	48.0	246	49.1
	549	83.7	70	82.4	5385	46.9	286	48.4
CapsE	–	–	–	–	719	56.0	303	59.3
	244	93.9	72	88.3	711	56.2	301	59.4
ConvKB	–	–	–	–	763	56.7	254	53.2
	204	94.7	66	88.7	759	56.8	275	54.7
PConvKB (local)	212	95.3	58	89.6	733	57.0	267	57.5
PConvKB (global)	249	93.8	63	89.1	749	56.8	283	56.2
PConvKB	**196**	**96.3**	**54**	**91.4**	**691**	**57.4**	**245**	**59.8**

Results. Table 2 shows the link prediction results of our model and the comparison methods on the four benchmark datasets. From the results, we can observe that:

1. PConvKB obtains the best MR and highest Hits@10 scores on the four benchmark datasets, demonstrating the effectiveness of incorporating relation paths for knowledge graph embedding.
2. Among PConvKB, PConvKB (local) and PConvKB (global), PConvKB obtains the best performance, which indicates that considering relation paths locally and globally is beneficial for knowledge graph embedding.
3. PConvKB does better than the closely related model ConvKB on all experimental datasets, especially on FB15k where PConvKB gains significant improvements of $275 - 247 = 28$ in MR (which is about 10.1% relative improvement) and $59.8\% - 54.7\% = 5.1\%$ absolute improvement in Hits@10.

4.4 Triple Classification

Triple classification task is to determine whether a given triple (h, r, t) is correct or not, i.e., binary classification on a triple.

Evaluation Protocol. We follow the same protocol in [33]. For each triple in test set and validation set, we construct one negative triple by switching entities from test triples and validation triples, respectively. The triple classification decision rule is: for a triple (h, r, t), if its implausibility score is below the relation-specific threshold σ_r, predict positive, otherwise negative. The relation-specific threshold σ_r is determined by maximizing classification accuracy on the validation set. The triple classification accuracy is the percentage of triples in the test set that are classified correctly.

Implementation Details. We use TransE to initialize entity and relation embeddings in PConvKB, select the learning rate in $\{5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}\}$, the number of filters in $\{50, 100, 200, 400\}$. We set the batch size at 128 and set the L_2-regularizer λ at 0.001 in our objective function. We run PConvKB up to 150 epochs and monitor the accuracy after every 10 training epochs to choose optimal hyper-parameters. We obtain the highest accuracy on the validation set when learning rate at $5e^{-5}$, the number of filters at 400 on WN18; and learning rate at $1e^{-5}$, the number of filters at 50 on FB15k; and the learning rate at $5e^{-6}$, the number of filters at 400 on WN18RR; and the learning rate at $1e^{-5}$, the number of filters at 200 on FB15k-237. For comparison methods, we implement them by the codes released by [11], [7] and [22].

Results. Table 3 shows the triple classification results of our model and the comparison methods on the four benchmark datasets. From the results, we can observe that:

Table 3. Experiments results on triple classification (%). The best score is in **bold**, while the second best score is in underline.

Model	WN18	FB15k	WN18RR	FB15k-237
TransE	87.6	82.9	74.0	75.6
TransH	96.5	85.7	77.0	77.0
TransD	96.4	86.1	76.3	77.0
HolE	88.1	82.6	71.4	70.3
ConvE	95.4	87.3	78.3	78.2
CapsE	96.5	<u>88.4</u>	79.6	79.5
ConvKB	96.4	87.9	79.1	80.1
PConvKB (local)	<u>97.5</u>	88.1	<u>79.7</u>	80.6
PConvKB (global)	96.9	87.6	79.4	<u>80.9</u>
PConvKB	**97.6**	**89.5**	**80.3**	**82.1**

1. On the whole, PConvKB yields the best performance on the four benchmark datasets, which is consistent with the results of link prediction, and further illustrates taking the relation paths into consideration is beneficial for knowledge graph embedding.
2. More specifically, on FB15k-237, the accuracy of triple classification improves from 80.6% of PConvKB(locally) to 82.1% PConvKB, and 80.9% of PConvKB (global) to 82.1% PConvKB. It demonstrates that considering the importances of relation paths locally and globally can better improve the knowledge graph embedding.

5 Related Work

Various methods have been proposed for knowledge graph embedding, such as general linear-based models [6], bilinear-based models [13,27,34], translation-based models [5,9,14,15,20,41,43], and neural network-based models [4,7,22,23,31–33]. We refer to [24,39] for a recent survey. In this section, we focus on the most relevant neural network-based models, and briefly review the other related methods.

Socher et al. [33] introduce neural tensor networks for knowledge graph embedding, which allows mediated interaction of entity embeddings via a tensor. Schlichtkrull et al. [31] present relational graph convolutional networks for knowledge graph completion. Shi and Weninger [32] present a shared variable neural network model called ProjE, which fills-in missing facts in a knowledge graph by learning joint embeddings of entities and relations. Dettmers et al. [7] present a multi-layer convolutional network model, namely ConvE, which uses 2D convolutions over embeddings to predict missing links in knowledge graphs. Nguyen et al. [22] present a CNN-based embedding model, i.e., ConvKB. It applies CNN to explore the global relationships among same dimensional entries in each embedding triple, which generalizes the transitional characteristics in the transition-based embedding models. Nguyen et al. [23] present CapsE, which combines CNN with capsule networks [29] for knowledge graph embedding. All these models treat a knowledge graph as a collection of triples, and disregard the rich information exist in relation paths.

There are several translation-based models [12,19,37,42,44] incorporating relation paths to improve the embeddings of entities and relations. However, they fully rely on hand-designed features to measure the importance of each path, which is not differentiable and cannot adjust during training. Moreover, they all based on translation-based models, which are not suitable for CNN-based model. To the best of our knowledge, our model PConvKB is the first attempt which incorporates relation paths in CNN-based embedding model.

6 Conclusion

In this paper, we present a novel CNN-based embedding model PConvKB, which improves knowledge graph embedding by incorporating relation paths locally and globally. In particular, we introduce attention mechanism to measure the local importance of relation paths. Moreover, we propose a simple yet effective measure DIPF to compute the global importance of relation paths. We evaluate our model on link prediction and triple classification. Experimental results show that our model achieves substantial improvements against state-of-the-art methods.

Acknowledgments. We acknowledge anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (Grant No. 61872045), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61921003).

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015 (2015). http://arxiv.org/abs/1409.0473
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
4. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Mach. Learn. **94**(2), 233–259 (2014). https://doi.org/10.1007/s10994-013-5363-6

5. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
6. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, 7–11 August 2011 (2011). http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659
7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
8. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610. ACM (2014)
9. Ebisu, T., Ichise, R.: TorusE: knowledge graph embedding on a lie group. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-2018), the 30th Innovative Applications of Artificial Intelligence (IAAI-2018), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-2018), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 1819–1826 (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16227
10. Ghosh, S., Desarkar, M.S.: Class specific TF-IDF boosting for short-text classification: application to short-texts generated during disasters. In: Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon, France, 23–27 April 2018, pp. 1629–1637 (2018). https://doi.org/10.1145/3184558.3191621
11. Han, X., et al.: OpenKE: an open toolkit for knowledge embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018, pp. 139–144 (2018). https://aclanthology.info/papers/D18-2024/d18-2024
12. Huang, W., Li, G., Jin, Z.: Improved knowledge base completion by the path-augmented TransR model. In: Li, G., Ge, Y., Zhang, Z., Jin, Z., Blumenstein, M. (eds.) KSEM 2017. LNCS (LNAI), vol. 10412, pp. 149–159. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63558-3_13
13. Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held at 3–6 December 2012, Lake Tahoe, Nevada, United States, pp. 3176–3184 (2012). http://papers.nips.cc/paper/4744-a-latent-factor-model-for-highly-multi-relational-data
14. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: 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 (Volume 1: Long Papers), vol. 1, pp. 687–696 (2015)
15. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, 12–17 February 2016, pp. 985–991 (2016). http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11982
16. Kim, D., Seo, D., Cho, S., Kang, P.: Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. Inf. Sci. **477**, 15–29 (2019). https://doi.org/10.1016/j.ins.2018.10.006

17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015 (2015). http://arxiv.org/abs/1412.6980
18. Li, X., et al.: Beyond RNNs: positional self-attention with co-attention for video question answering. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 8658–8665 (2019). https://aaai.org/ojs/index.php/AAAI/article/view/4887
19. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 705–714 (2015). http://aclweb.org/anthology/D/D15/D15-1082.pdf
20. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
21. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
22. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: 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, Volume 2 (Short Papers), vol. 2, pp. 327–333 (2018)
23. Nguyen, D.Q., Vu, T., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A capsule network-based embedding model for knowledge graph completion and search personalization. arXiv preprint arXiv:1808.04122 (2018)
24. Nguyen, D.Q.: An overview of embedding models of entities and relationships for knowledge base completion. CoRR abs/1703.08098 (2017). http://arxiv.org/abs/1703.08098
25. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: STransE: a novel embedding model of entities and relationships in knowledge bases. In: NAACL HLT 2016: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, 12–17 June 2016, pp. 460–466 (2016). https://www.aclweb.org/anthology/N16-1054/
26. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
27. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–2 July 2011, pp. 809–816 (2011). https://icml.cc/2011/papers/438_icmlpaper.pdf
28. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A Meeting of SIGDAT, a Special Interest Group of the ACL, Doha, Qatar, 25–29 October 2014, pp. 1532–1543 (2014). https://www.aclweb.org/anthology/D14-1162/

29. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 3856–3866 (2017). http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules
30. Saeedan, F., Weber, N., Goesele, M., Roth, S.: Detail-preserving pooling in deep networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018, pp. 9108–9116 (2018). http://openaccess.thecvf.com/content_cvpr_2018/html/Saeedan_Detail-Preserving_Pooling_in_CVPR_2018_paper.html
31. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
32. Shi, B., Weninger, T.: ProjE: embedding projection for knowledge graph completion. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017, pp. 1236–1242 (2017). http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14279
33. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held at 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 926–934 (2013). http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion
34. Sutskever, I., Salakhutdinov, R., Tenenbaum, J.B.: Modelling relational data using Bayesian clustered tensor factorization. In: Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a Meeting Held at 7–10 December 2009, Vancouver, British Columbia, Canada, pp. 1821–1828 (2009). http://papers.nips.cc/paper/3863-modelling-relational-data-using-bayesian-clustered-tensor-factorization
35. Tong, Z., Tanaka, G.: Hybrid pooling for enhancement of generalization ability in deep convolutional neural networks. Neurocomputing **333**, 76–85 (2019). https://doi.org/10.1016/j.neucom.2018.12.036
36. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 57–66 (2015)
37. Toutanova, K., Lin, V., Yih, W., Poon, H., Quirk, C.: Compositional learning of embeddings for relation paths in knowledge base and text. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Volume 1: Long Papers, Berlin, Germany, 7–12 August 2016 (2016). http://aclweb.org/anthology/P/P16/P16-1136.pdf
38. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledge base. Commun. ACM **57**, 78–85 (2014)
39. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499

40. Wang, W., Chen, Z., Hu, H.: Hierarchical attention network for image captioning. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 8957–8964 (2019). https://aaai.org/ojs/index.php/AAAI/article/view/4924
41. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
42. Xiong, S., Huang, W., Duan, P.: Knowledge graph embedding via relation paths and dynamic mapping matrix. In: Woo, C., Lu, J., Li, Z., Ling, T.W., Li, G., Lee, M.L. (eds.) ER 2018. LNCS, vol. 11158, pp. 106–118. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01391-2_18
43. Yuan, J., Gao, N., Xiang, J.: TransGate: knowledge graph embedding with shared gate structure. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 3100–3107 (2019). https://aaai.org/ojs/index.php/AAAI/article/view/4169
44. Zhang, M., Wang, Q., Xu, W., Li, W., Sun, S.: Discriminative path-based knowledge graph embedding for precise link prediction. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) ECIR 2018. LNCS, vol. 10772, pp. 276–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76941-7_21