Journal of Computers Vol. 30 No. 6, 2019, pp. 207-215
doi:10.3966/199115992019123006016

Knowledge Representation Learning with Dynamic Path

Yiqin Luo, Liang Chang*, Long Li, Wei Chen, Tianlong Gu

Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
lyq_ah@163.com, changl@guet.edu.cn

Received 13 January 2019; Revised 21 January 2019; Accepted 12 March 2019

Abstract. Representation learning aims to embed knowledge graphs into a low-dimensional, dense, and real-valued vector space, in which entities and relations in a knowledge graph are represented as vectors. Many models have been proposed in the literature for the embedding. A model will perform better if it can capture more information than other models during the embedding process. Compared with the classical model TransE, the model PTransE takes into account not only direct relations but also multi-step relations (i.e, paths) between each pair of entities, and thereby achieves significant improvement in the tasks of entity prediction and relation prediction. However, in the case that there are many multi-step relations between a pair of entities, PTransE doesn't make any distinguish between them. In this paper, by introducing dynamic factors into the path embedding process of the PTransE model, we propose a dynamic path translation (DPT) method to capture different paths between each pair of entities. Experimental results show that the DPT method has a significant improvement in the entity prediction task and the relation prediction task.

Keywords: dynamic translation, knowledge graph, multi-step relations, representation learning

1 Introduction

Knowledge graph is one of the most popular approaches for representing knowledge. Large-scaled knowledge graph supports many downstream natural language processing tasks, like question answering [1-2], page storing [3], etc. Many vertical industries, including education, finance and health care, are trying to introduce knowledge graph to big data analysis. However, existing knowledge graph has a big flaw: incompleteness. It greatly limits the application of knowledge graph. Therefore, how to obtain high-quality knowledge representation and use the semantic information of the knowledge graph for knowledge reasoning has received extensive attention.

The earliest knowledge reasoning is carried out by inducing some logic based reasoning rules [4-5], like FOIL [6]. Wang et al. forming the concept of ontology for reasoning [7].

Graph-based reasoning is mainly based on multi-step link prediction to solve the problem of automatic reasoning. On the graph, this method starts from a source entity and randomly walk to another entity. If it can reach the target entity, there is a connection between the source entity and the target entity. Some earlier works like PRA [8] use bounded-depth random walk with restarts to obtain paths. More recently, DIVA [9] uses variation inference to maximize the evidence lower bound.

Vector-space-based reasoning aims to embed entities and relations into a low-dimensional dense real-valued vector space to gain their vector representations. This method infers missing knowledge through vector calculations, which is called knowledge representation learning. There are several typical models, such as structured embedding (SE), single layer model (SLM), semantic matching energy (SME), latent factor model (LFM), neural tensor network, RESCAL and translation model, etc. Among them, TransE [10] model (translation model) proposed by Bordes et al. in 2013 is most effective. It is based on the unbalanced phenomenon of word vector space proposed by Mikolov. In recent years, many extension models based on TransE have been proposed. In order to solve the problems encountered by TransE in

* Corresponding Author

dealing with complex relations, the TransR [11] model projects each triple to the corresponding relation space, and then establishes the translation between the head and tail entities in the relation space. The DT [12] model proposes a flexible translation principle that can better handle the problem of modeling complex relations and entities. These models have achieved good results, but they only consider the direct relation between entities. In fact, there are abundant semantic information in multi-step relations between entity pairs. These continuous multi-step relations from the head entity to the tail entity are called paths. Knowledge reasoning is not limited to single step reasoning of modeling direct relations, and multi-step reasoning of modeling paths is obtaining more and more attention. Based on paths, the researchers have successfully proposed a series of models like PTransE [13], RTransE [14], and PTransR [15]. Existing path-based representation learning models have achieved significant improvement. However, when there are more than one relation or path between entity pairs, these models cannot represent their semantic information well.

Existing optimization methods are too strict to distinguish several paths of an entity pair. In order to solve the problem of modeling several paths of an entity pair, we propose a dynamic translation representation learning model based on the path. In the model, dynamic factors are added to paths representation process to achieve flexible transformation between relations and paths embedding. Thus, this method can effectively learning similar paths to alleviate the above problem. Meanwhile, it is more suitable for complex reasoning patterns in large-scale knowledge graphs than existing methods.

The rest of this paper is organized as follows. We first give an introduction to the TransE model and PTransE model in Section 2, and then present the DPT method in section 3. In Section 4 we evaluate the performance of DPT method by comparing it with many existing models on the task of entity prediction and relation prediction. Section 5 concludes the paper.

2 Basic Model

Several symbols are described as follows. The knowledge base is denoted as $G=(E, R, S)$. $E=(e_1, e_2, \cdots, e_{|E|})$ is a set of entities in the knowledge base, which contains $|E|$ different entities; $R=(r_1, r_2, \cdots, r_{|R|})$ is a set of relations in the knowledge base, which contains $|R|$ kinds of different relations; $S \subseteq E \times R \times E$ represents a set of triples in the knowledge base, expressed as (h, r, t), where h and t represent head entities and tail entities respectively, and r represents the relation between h and t. For example, the triple (*Michael Jordan*, *Nationality*, *America*) indicates that there is a relation "Nationality" between the entities "Michael Jordan" and "America". $p=\{r_1, r_2, \cdots, r_l\}$ is a path between two entities of a given entity pair, denoted as $h \xrightarrow{r_1} \ldots \xrightarrow{r_l} t$, and l is the number of relations. When l=2, p is named 2-step path. Likewise, when l=3, p is named 3-step path. $P(h,t)=\{p_1,\ldots,p_N\}$ is a set of paths between h and t, and N is the number of relation paths. For example, the relation triple (h, *Nationality*, t) can be represented as $h \xrightarrow{Nationality} t$, and the path triple ($h$, { *BornInCity*, *CityInState*, *StateInCountry* }, t) (short for (h, p, t))is represented as $h \xrightarrow{BornInCity} e_1 \xrightarrow{CityInState} e_2 \xrightarrow{StateInCountry} t$, which contains three triples (h, *BornInCity*, e_1), (e_1, *CityInState*, e_2) and (e_2, *StateInCountry*, t), and p={ *BornInCity*, *CityInState*, *StateInCountry* }.

2.1 TransE Model

TransE considers the relation r as a translation between the head entity h and tail entity t for each triple (h,r,t). And Fig. 1 shows a diagram of TransE model. The basic idea of TransE is to embed all the entities and relations in the knowledge graph into the same low-dimensional vector space. Then, we can get the head entity vector **h**, the tail entity vector **t**, and the relation vector **r** between the head entity and the tail entity. A gradient descent algorithm is used to minimize the loss function to reduce the value of the score function so as to achieve the goal of **h**+**r**=**t**. The score function for training the embedded vector is defined as:

$$E(h,r,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}. \quad \textbf{(1)}$$

where L_1 denotes a L_1-norm, L_2 denotes a L_2-norm.

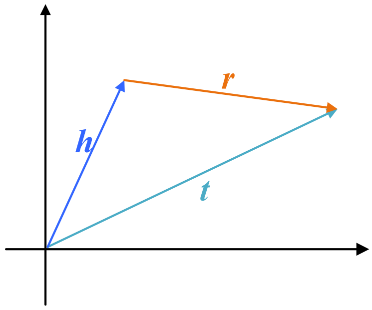
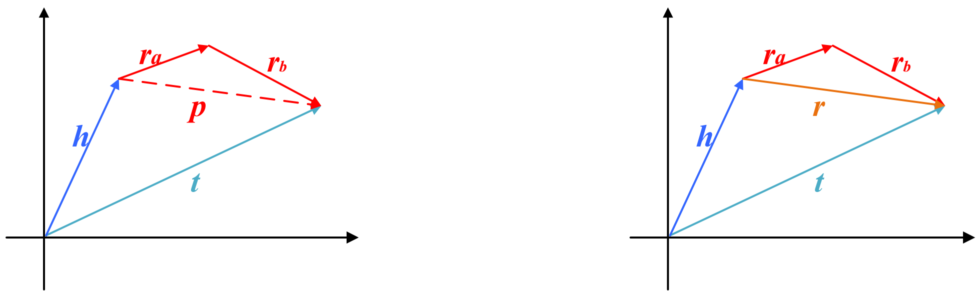


Fig. 1. Simple illustration of TransE

2.2 PTransE Model

TransE only consider direct relations but ignore multi-step relation paths. Actually, paths also represent semantic information between entities. To alleviate the limitation of the TransE model for isolated learning of each triple, the PTransE model adds a path based on the TransE model as shown in Fig. 2(b). It considers the path *p* as a translation from the head entity *h* to the tail entity *t*. Then the path triple (*h*, *p*, *t*) can be obtained. Due to paths are to be considered, it should also be represented in low-dimensional dense vector spaces. Since the semantic information of a path depends on all the relations on this path, the model defines and learns a binary operation function, which forms the embedding of the path *p* by recursion. Here is an example of addition operation, as shown in Fig. 2(a). From this operation, we can get the path vector **p** between the head entity and the tail entity. But not all paths are meaningful and reliable, so the author proposes a path-constraint resource allocation (PCRA) algorithm to measure the reliability of paths.



(a) (b)

Fig. 2. Simple illustration of PTransE

For the triple (*h*, *r*, *t*) of a given entity pair (*h*, *t*), TransE considers the relation vector **r** as the translation vector between the entity vectors **h** and **t**, and its energy function is equal to formula (1). PTransE considers not only the relation *r* but also the path *p*, whose energy function is defined as:

$$G(h,r,t) = E(h,r,t) + E(h,P,t)\,. \tag{2}$$

where $E(h,P,t)$ is the statistics of all the path for a given entity pair, it defined as:

$$E(h,P,t) = \frac{1}{Z}\sum_{p\in P(h,t)} R(p|h,t)E(h,p,t)\,. \tag{3}$$

$$E(h,p,t) = \|\mathbf{h} + \mathbf{p} - \mathbf{t}\|_{L_1/L_2}\,. \tag{4}$$

$R(p|h,t)$ represents the reliability of the path between a given entity pairs (*h*, *t*), and its value is calculated by the PCRA algorithm. $Z = \sum_{p\in P(h,t)} R(p|h,t)$ is a normalization factor.

3 Dynamic Path Translation

3.1 Motivation

In the path-based representation learning model, relation r in the relation triple (h, r, t) is considered to represent the direct relationship between two entities. Moreover, path p in the path triple (h, p, t) represents the indirect connection of an entity pair. When the relation r and the path p connect the same entity pair (h, t), there are two types of triple (h, r, t) and (h, p, t). Then the entity pair has to satisfy score functions $E(h,r,t)=\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{L_1/L_2}$ and $E(h,p,t)=\|\mathbf{h}+\mathbf{p}-\mathbf{t}\|_{L_1/L_2}$. Since the relation r and path p are connected to the same entity pair (h, t), we can get an equation **r**=**p**, that is, **r-p=0**. Existing path-based representation learning models use **r-p=0** as the translation principle. These models do well in situation that an entity pair only contains one relation and one path. However, there are some issues when the number of relation or path is more than one for an entity pair. As shown in Fig. 3, given entity pair (h, t) contains relations $R=\{r_1,r_2\}$ and paths $P=\{p_1,p_2\}$ where $p_1=\{r_{11},r_{12}\}$ and $p_2=\{r_{21},r_{22}\}$. Then we can get $\mathbf{p_1}=\mathbf{r_1}=\mathbf{p_2}=\mathbf{r_2}$. But these vectors are absolutely equal only when representing the same semantic information.

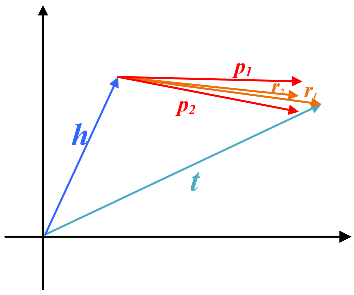
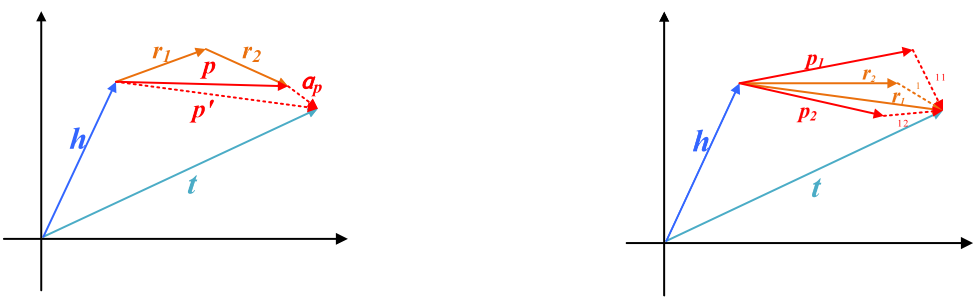


Fig. 3. Illustration of the DPT method

In order to solve this problem and improve training efficiency, we propose a new dynamic path translation (DPT) principle, and its basic method is shown in the Fig. 4. Specifically, when (h, t) exists, a dynamic factor, $\boldsymbol{\alpha}$ vector is added to the path during its presentation. During the training, the path vector is optimized to a target vector which is very close to the direct relation vector. As long as the path vector is in a certain range, it will be considered to have expressed the semantic information without having to be strictly equal to the target vector. This principle of translation is more tolerant and allows certain errors within a small range. Therefore, we can get $\mathbf{p_1}+\boldsymbol{\alpha}_{11}=\mathbf{r_1}$, $\mathbf{p_1}+\boldsymbol{\alpha}_{11}=\mathbf{r_1}$, $\mathbf{p_1}+\boldsymbol{\alpha}_{21}=\mathbf{r_2}$ and $\mathbf{p_2}+\boldsymbol{\alpha}_{22}=\mathbf{r_2}$. Since each of the factors $\boldsymbol{\alpha}$ is dynamically generated and not equal to each other, that is to say, $\boldsymbol{\alpha}_{11}\neq\boldsymbol{\alpha}_{12}\neq\boldsymbol{\alpha}_{12}\neq\boldsymbol{\alpha}_{12}$, so $\mathbf{p_1}\neq\mathbf{r_1}\neq\mathbf{p_2}\neq\mathbf{r_2}$. The factor $\boldsymbol{\alpha}$ is small enough, so it cannot affect the expression of similar semantic information. Unlike previous models, our translation principle is more flexible and fits the facts. In this way, similar paths can be learned simply and efficiently.



(a) (b)

Fig. 4. Simple illustration of DPT

3.2 The DPT Model

We call the improved model of PTransE as DPT model. Scoring functions of DPT are similar with the scoring functions of PTransE, as shown by formula (2) and (3), Except that here $E(h,p,t)$ is defined as:

$$E(h,p,t) = \|\mathbf{h} - (\mathbf{p} + \boldsymbol{\alpha}) - \mathbf{t}\|_{L_1/L_2} . \quad \textbf{(5)}$$

where $\boldsymbol{\alpha}$ represents the dynamic factor, and $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{p}, \boldsymbol{\alpha} \in R^n$.

The loss function adopted by DPT is as follows:

$$L(S) = \sum_{(h,r,t)\in S} [L(h,r,t) + \frac{1}{Z} \sum_{p\in P(h,t)} R(p|h,t) L(p,r)] . \quad \textbf{(6)}$$

where $L(h,r,t)$ and $L(p,r)$ represent the loss values of triple (h,r,t) and (h,p,t) respectively.

$$L(h,r,t) = \sum_{(h',r',t')\in S^-} [\gamma + E(h,r,t) - E(h',r',t')]_+ . \quad \textbf{(7)}$$

and

$$L(p,r) = \sum_{(h,r',t)\in S^-} [\gamma + E(p,r) - E(p,r')]_+ . \quad \textbf{(8)}$$

where $[x]_+ = \max(0; x)$ returns the maximum one between 0 and x, γ is the margin, S is the set of valid triples existing in KB, S^- is the set of invalid triples. Compared with the invalid triples, the optimization goal will help to reduce the score of the valid triples.

According to the triple (h,r,t), invalid triples are defined as:

$$S^- = \{h',r,t\} \cup \{h,r',t\} \cup \{h,r,t'\} . \quad \textbf{(9)}$$

S^- can be obtained by replacing h, r, or t in (h,r,t), which contains three situations:

(1) replace the head entity randomly:

$$E(h',r',t') = ||\mathbf{h}' + \mathbf{r} \text{ - } \mathbf{t}||_{L_1/L_2} . \quad \textbf{(10)}$$

(2) replace the relation randomly:

$$E(h',r',t') = ||\mathbf{h} + \mathbf{r}' \text{ - } \mathbf{t}||_{L_1/L_2} . \quad \textbf{(11)}$$

(3) replace the tail entity randomly:

$$E(h',r',t') = ||\mathbf{h} + \mathbf{r} \text{ - } \mathbf{t}'||_{L_1/L_2} . \quad \textbf{(12)}$$

In the above equations, $\mathbf{h}'$ represents the random replacement of the head entity vector, $\mathbf{r}'$ represents the random replacement of the relation vector, and $\mathbf{t}'$ represents the random replacement of the tail entity vector. For the same head entity and the same tail entity, due to $E(h,p,t) = \|(p+\alpha) - (t-h)\|_{L_1/L_2}$ $= \|(p+\alpha) - r\|_{L_1/L_2} = E(p,r)$, the positive relation triple corresponding to the valid path triple scoring function is:

$$E(p,r) = \|(\mathbf{p} + \boldsymbol{\alpha}) - \mathbf{r}\|_{L_1/L_2} . \quad \textbf{(13)}$$

the invalid relation triple corresponding to the valid path triple scoring function is:

$$E(p,r') = \|(\mathbf{p} + \boldsymbol{\alpha}) - \mathbf{r}'\|_{L_1/L_2} . \quad \textbf{(14)}$$

4 Evaluation

4.1 Data Sets and Experiment Settings

A typical large-scale knowledge graph Freebase [16] is used to evaluate our method. In this paper, we adopt FB15K [17] a subset of Freebase for entity prediction and relation prediction tasks. The statistics of the data sets are listed in Table.1.

Table 1. Statistics of the FB15K dataset

Dataset	#Rel	#Ent	#Train	#Valid	#Test
FB15K	1,345	14,951	483,142	50,000	59,071

In Table 2, we list the complexity of partial models introduced in introduction, and compare it with our model. Among them, N_e and N_r respectively represent the number of entities and relations, N_t represents the number of triples in a knowledge graph, m denotes the dimension of the entity's embedded space, n denotes the dimension of the relation embedded space, k is regarded as the number of nodes implied by a neural network, s is regarded as the number of slices of the tensor, P is the expected number of paths between two entities, and L is the expected path length.

Table 2. Complexities (the number of parameters and the times of operations) of several embedding models

Model	#Parameters	#Operations(Time complexity)
SLM (Socher et al. 2013)	$O(N_e m+N_r(2k+2nk))(m=n)$	$O((2mk+k)N_t)$
NTN (Socher et al. 2013)	$O(N_e m+N_r(n^2 s+2ns+2s))(m=n)$	$O(((m^2+m)s+2mk+k)N_t)$
TransE (Bordes et al. 2013)	$O(N_e m+2N_r n)(m=n)$	$O(N_t)$
TransH (Wang et al. 2014)	$O(N_e m+2N_r n)(m=n)$	$O(2mN_t)$
TransR (Lin et al.2015)	$O(N_e m+N_r(m+1)n)(m=n)$	$O(2mnN_t)$
PTransE (Bordes et al. 2015)	$O(N_e m+2N_r n)(m=n)$	$O(N_t PL)$
PTransR (Bordes et al. 2017)	$O(N_e m+N_r(m+1)n)(m=n)$	$O(2mnN_t PL)$
DPT	$O(N_e m+2N_r n)(m=n)$	$O(3mN_t PL)$

4.2 Entity Prediction

In entity prediction task, our work is to predict missing entities h or t in the relation triple (h,r,t) [10]. In this task, for each missing entity, the system gives a set of candidate entities from the knowledge graph, rather than just giving a best result. As described in [10, 17], the data set FB15k was used to perform experiments in the entity prediction task.

As with [10], we use two methods as evaluation criteria: (1) the average ranking of the correct entities (Mean Rank), (2) the probability of the correct entities in the top 10 (Hits@10). When the lower the average ranking is and the higher the Hits@10 is, the result is better. In fact, there may be some invalid triples in the knowledge graph. During training Process, the above evaluation method may consider such triples are correct and treat them as valid triples. It may lead to that these invalid triples have a higher ranking than the original valid triples. Therefore, before ranking, these invalid triples that appear in the knowledge graph need to be removed first. Here we set the first evaluation to "Raw" and the latter to "Filter". This paper will show the results of these two settings.

It is time-consuming and impractical to find all possible relationships between a given entity and each candidate entity. In this paper, we adopt a re-ranking method proposed by PTransE. This method uses TransE to obtain top-500 candidates after ranking and then uses PTransE to re-rank this top-500 candidates. After testing a various number of paramater of model, we empirically choose the best configuration on FB15k is λ=0.001, γ=1, k=100, d=100, B=1440. And we adopt L_1 as the similarity measure distance.

The experimental results on FB15k are shown in Table 3. The overall effect of DPT is ideal. More specifically, (1) For Hits@10 "Filter", DPT's result is the best compared with all current path models. Even compared with PTransE, the ranking increased from 83.4% to 87.4% in 2-step, 84.6% to 85.4% in 3-step. (2) For Hits@10 "Raw", DPT is basically the same as other models. (3) For Mean Rank, DPT performs better than most of current models. Although there is a certain gap compared with PTransR, the time complexity of DPT is significantly lower.

Table 3. Evaluation results of entity prediction on FB15K

Metric	Hits@10 (%)		Mean Rank	
	Raw	Filter	Raw	Filter
RESCAL	28.4	44.1	828	683
SE	28.8	39.8	273	162
SME (linear)	30.7	40.8	274	154
SME (bilinear)	31.3	41.3	284	158
LFM	26.0	33.1	284	164
TransE	34.9	47.1	243	125
TransH	45.7	64.4	212	87
TransR	48.2	68.7	198	77
PTransE (2-step)	51.8	83.4	200	54
PTransE (3-step)	51.4	84.6	207	58
PTransR (2-step)	**53.0**	84.3	**171**	**47**
DPT (2-step)	52.9	**87.4**	212	61
DPT (3-step)	49.6	85.4	217	62

For Hits@10 "Filter", we conducted further research and analysis. The results are shown in Table 4. (1) For N-to-N, DPT is superior to all current path models, which can obtain relatively ideal performance in the terms of predicting head and tail entities. (2) For 1-to-1, DPT is basically the same as the current optimal results. (3) For 1-to-N and N-to-1, DPT's results are better than all current models when predicting 1-terminus (head entities in 1-to-N or tail entities in N-to-1), and are second only to PTransR when predicting the N-terminus (head entities in N-to-1 or tail entities in 1-to-N).

Table 4. Evaluation results of different relation catogories

Tasks	Predicting Head Entities Hits@10 (%)				Predicting Tail Entities Hits@10 (%)			
Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
SE	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear)	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (bilinear)	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
PTransE (2-step)	91.0	92.8	60.9	83.8	91.2	74.0	88.9	86.4
PTransE (3-step)	90.1	92.0	58.7	86.1	90.7	70.7	87.5	88.7
PTransR	**91.4**	93.4	**65.5**	84.3	91.2	**74.5**	91.8	86.8
DPT (2-step)	90.9	**94.6**	60.9	**89.3**	91.0	70.0	**92.6**	**91.1**
DPT (3-step)	90.7	92.6	57.8	87.0	**91.3**	69.6	89.5	89.6

4.3 Relation Prediction

The relation prediction is intended to predict the relation between two given entities. This task is still evaluated on FB15K. In this sub-task, we can use the DPT's scoring function to rank the candidate relations, rather than re-ranging as the entity prediction. As can be seen from Table.5, the result of the Hits@10 "Filter" evaluation have made some improvements.

Table 5. Evaluation results on relation prediction

Metric	Mean Rank		Hits@1(%)	
	Raw	Filter	Raw	Filter
PTransE (2-step)	1.7	1.2	69.5	93.6
-TransE	135.8	135.3	51.4	78.0
-Path	2.0	1.6	69.7	89.0
PTransE (3-step)	1.8	1.4	68.5	94.0
DPT (2-step)	1.9	1.4	69.1	94.6
DPT (3-step)	2.0	1.5	65.9	94.9

5 Conclusion

In this paper, we propose a novel dynamic path translation (DPT) model. The basic idea of DPT is that different dynamic factors are added to different paths to obtain a translation principle. This principle has a certain capacity for fault tolerance and it makes our model more realistic. At the same time, this method makes our model more efficient to learn relations and paths between two entities. Compared with other existing models, our method improves the accuracy of experimental results without increasing the complexity, such as the Hits@10 proportions, especially in the N-to-N type relationship. Our model has improved the ability of complex knowledge reasoning. However, the model still has a little shortcoming, such as the slight improvement of Mean Rank. Due to the model has a certain fault tolerance, it is difficult to improve Hits@10 while improving Mean Rank. In the future, we will improve the Mean Rank without affecting Hits@10.

Acknowledgements

This work is supported by the Natural Science Foundation of China (Nos. 61572146, U1501252, U1711263); the Natural Science Foundation of Guangxi Province (No. 2016GXNSFDA380006); and the High Level of Innovation Team of Colleges and Universities in Guangxi and Outstanding Scholars Program.

References

[1] J.-W. Bao, N. Duan, Z. Yan, M. Zhou, T.-J. Zhao, Constraint-based question answering with knowledge graph, COLING (2016) 2503-2514.

[2] W.-Y. Cui, Y.-H. Xiao, H.-X. Wang, Y.-Q. Song, S.-W. Hwang, W. Wang, KBQA: learning question answering over QA corpora and knowledge bases, PVLDB 10(5)(2017) 565-576.

[3] Y.-J. Du, Q. Hu, X.-L. Li, X.-L. Chen, C.-X. Li, Ranking web page with path trust knowledge graph, IScIDE (2)(2015) 66-75.

[4] J. McCarthy, Programs with Common Sense, RLE and MIT Computation Center, Cambridge, 1960.

[5] N.J. Nilsson, Logic and artificial intelligence, Artificial intelligence 47(1991) 31-56.

[6] S. Muggleton, Inverse entailment and progol, New Generation Computing 13(1995) 245-286.

[7] C.-L. Wang, X.-W. Zhang, Z.-Y. Feng, H.-W. Qin, A graph-based approach to incremental classification in OWL 2 QL Ontology, in: J. Li, M. Zhou, G. Qi, N. Lao, T. Ruan, J. Du (Eds.), Knowledge Graph and Semantic Computing. Language, Knowledge, and Intelligence, Springer, Singapore, 2017, pp. 18-29.

[8] N. Lao, T.M. Mitchell, W.W. Cohen, Random walk inference and learning in a large scale knowledge base, in: Proc. EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011.

[9] W.-H. Chen, W.-H. Xiong, X.-F Yan, W.Y. Wang, Variational knowledge graph reasoning, in: Proc. NAACL-HLT 2018, 2018.

[10] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnen-ko, Translating embeddings for modeling multi-relational data, in: Proc. Neural Information Processing Systems 2013, 2013.

[11] Y.-K. Lin, Z.-Y. Liu, M.-S. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proc. AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[12] L. Chang, M.-L. Zhu, T.-L. Gu, C.-Z Bin, J.-Y Qian, J. Zhang, Knowledge graph embedding by dynamic translation, IEEE Access 5(2017) 20898-20907.

[13] Y.-K. Lin, Z.-Y. Liu, H.-B. Luan, M.-S. Sun, S.-W. Rao, S. Liu, Modeling relation paths for representation learning of knowledge bases, in: Proc. the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[14] A. García-Durán, A. Bordes, N. Usunier, Composing relationships with translations, in: Proc. the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[15] W.-H. Huang, G. Li, Z. Jin, Improved knowledge base completion by the path-augmented transR model, in: G. Li, Y. Ge, Z. Zhang, Z. Jin, M. Blumenstein (Eds.), KSEM 2017, Springer, Cham, pp. 149-159.

[16] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in: Proc. SIGMOD '08 Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2018.

[17] A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data, Machine Learning 94(2)(2014) 233-159.