

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)  
<http://www.disi.unitn.it>

## **Relation path embedding in Knowledge graphs**

Xixun Lin, Yanchun Liang, Fausto  
Giunchiglia, Xiaoyue Feng, Renchu Guan

February 2018

Technical Report # DISI-18-011

Published in “Neural Computing and Applications  
Journal”, 2018



# Relation path embedding in knowledge graphs

Xixun Lin<sup>1</sup> · Yanchun Liang<sup>1,2</sup> · Fausto Giunchiglia<sup>3</sup> · Xiaoyue Feng<sup>1</sup> · Renchu Guan<sup>1,2</sup> 

Received: 8 August 2017 / Accepted: 21 February 2018  
© The Author(s) 2018. This article is an open access publication

## Abstract

Large-scale knowledge graphs have currently reached impressive sizes; however, they are still far from complete. In addition, most existing methods for knowledge graph completion only consider the direct links between entities, ignoring the vital impact of the semantics of relation paths. In this paper, we study the problem of how to better embed entities and relations of knowledge graphs into different low-dimensional spaces by taking full advantage of the additional semantics of relation paths and propose a novel relation path embedding model named as RPE. Specifically, with the corresponding relation and path projections, RPE can simultaneously embed each entity into two types of latent spaces. Moreover, type constraints are extended from traditional relation-specific type constraints to the proposed path-specific type constraints and both of the two type constraints can be seamlessly incorporated into RPE. The proposed model is evaluated on the benchmark tasks of link prediction and triple classification. The results of experiments demonstrate our method outperforms all baselines on both tasks. They indicate that our model is capable of catching the semantics of relation paths, which is significant for knowledge representation learning.

**Keywords** Knowledge graph completion · Relation paths · Path projection · Type constraints · Knowledge representation learning

## 1 Introduction

Large-scale knowledge graphs, such as Freebase [2], WordNet [22], Yago [28], and NELL [6], are critical to natural language processing applications, e.g., question answering [8], relation extraction [26], and language modeling [1]. These knowledge graphs generally contain billions of facts, and each fact is organized into a triple base format (head entity, relation, tail entity), abbreviated as  $(h, r, t)$ . However, the coverage of such knowledge graph is still far from complete compared with real-world

knowledge [9]. Traditional knowledge graph completion approaches, such as Markov logic networks [25], suffer from feature sparsity and low efficiency.

Recently, encoding the entire knowledge graph into a low-dimensional vector space to learn latent representations of entity and relation has attracted widespread attention. These knowledge embedding models yield better performance in terms of low complexity and high scalability compared with previous works. Among these methods, TransE [5] is a classical neural-based model, which assumes that each relation can be regarded as a translation from head to tail and uses a score function  $S(h, r, t)$  to measure the plausibility for triples. TransH [32] and TransR [20] are representative variants of TransE. These variants consider entities from multiple aspects, various relations may stress on different aspects.

However, the majority of these approaches only exploit direct links that connect head and tail entities to predict potential relations between entities. These approaches do not explore the fact that relation paths, which are denoted as the sequences of relations, i.e.,  $p = (r_1, r_2, \dots, r_m)$ , play an important role in knowledge inference. For example, the

✉ Renchu Guan  
guanrenchu@jlu.edu.cn

<sup>1</sup> Key Laboratory for Symbol Computation and Knowledge Engineering of National Education Ministry, College of Computer Science and Technology, Jilin University, Changchun 130012, China

<sup>2</sup> Zhuhai Laboratory of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Zhuhai College of Jilin University, Zhuhai 519041, China

<sup>3</sup> DISI, University of Trento, Trento, Italy

successive facts J.K. Rowling  $\xrightarrow{\text{CreatedRole}}$  Harry Potter  
 $\xrightarrow{\text{DescribedIn}}$  Harry Potter and the Philosopher's Stone can be used to infer the new triple (J.K. Rowling, WroteBook, Harry Potter and the Philosopher's Stone), which does not appear in the original knowledge graph. Consequently, using relation paths to learn meaningful knowledge embeddings is a promising new research direction [10, 12, 23, 29].

In this paper, we propose a novel relation path embedding model (RPE) to explicitly model knowledge graph by taking full advantage of the semantics of relation paths. By exploiting the compositional path projection, RPE can embed each entity into proposed path spaces for better tackling various relations with multi-mapping properties. Moreover, we extend the relation-specific type constraints [16] to the novel path-specific type constraints to improve the prediction quality. Our model is evaluated on three benchmark datasets with link prediction and triple classification. Experimental results show that RPE outperforms all baselines on these tasks, which proves the effectiveness of our method and the importance of relation paths.

The remainder of this paper is organized as follows. We first provide a brief review of state-of-the-art knowledge embedding models in Sect. 2. Two main intuitions of our model are illustrated in Sect. 3. The details of RPE are introduced in Sect. 4. The experiments and analyses are reported in Sect. 5. Conclusions and directions for future work are reported in the final section.

## 2 State-of-the-art

In this paper, We first concentrate on three classical translation-based models that only consider direct links between entities. The first translation-based model is TransE inspired by [21]. TransE defines the score function as  $S(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$  for each triple (the bold lowercase letter denotes a column vector). The score will become smaller if the triple  $(h, r, t)$  is correct; otherwise, the score will become higher. The embeddings are learned by optimizing a global margin-loss function. This assumption looks quite simple, but it was shown to achieve great results empirically. However, TransE cannot address more complex relations with multi-mapping properties well, i.e., 1-to-N (*bring out*), N-to-1 (*gender of*), and N-to-N (*work with*). To alleviate this problem, TransH projects entities into a relation-dependent hyperplane by the normal vector  $\mathbf{w}_r$ :  $\mathbf{h}_h = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r$  and  $\mathbf{t}_t = \mathbf{t} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r$  (restrict  $\|\mathbf{w}_r\|_2 = 1$ ). The corresponding score function is  $S(h, r, t) = \|\mathbf{h}_h + \mathbf{r} - \mathbf{t}_t\|$ . TransE and TransH achieve translations on the same embedding space, whereas TransR

assumes that each relation should be used to project entities into different relation-specific embedding spaces since different relations may place emphasis on different entity aspects. The projected entity vectors are  $\mathbf{h}_r = \mathbf{M}_r \mathbf{h}$  and  $\mathbf{t}_r = \mathbf{M}_r \mathbf{t}$  (the bold uppercase letter  $\mathbf{M}$  denotes a matrix); thus, the new score function is defined as  $S(h, r, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|$ . In addition, there are other novel knowledge embedding models [14, 15] focusing on efficient relation projection are evaluated in our experiments. Multi-source information learning for knowledge graph completion, which aims at utilizing heterogeneous data to enrich knowledge embedding, is a promising research. For instance, [33] employs the convolutional neural networks (CNN) to learn embeddings from extra entity description for zero-shot scenarios.

Another research direction focuses on improving the prediction performance by using prior knowledge in the form of relation-specific type constraints [7, 16, 30]. Note that each relation should contain *Domain* and *Range* fields to indicate the subject and object types, respectively. For example, the relation *HasChildren's Domain* and *Range* types both belong to a person. By exploiting these limited rules, the harmful influence of a merely data-driven pattern can be avoided in part. For example, Type-constrained TransE [16] imposes these constraints on the global margin-loss function to better distinguish similar embeddings in latent space.

A third related direction is PTransE [19] and the path ranking algorithm (PRA) [17]. PTransE considers relation paths as translations between head and tail entities and primarily addresses two problems: (1) exploits a variant of PRA to select reliable relation paths and (2) explores three path representations by compositions of relation embeddings. PRA, as one of the promising research innovations for knowledge graph completion, has also attracted considerable attention [11, 18, 24, 31]. It uses the path-constrained random walk probabilities as path features to train linear classifiers for different relations. In large-scale knowledge graphs, relation paths have great significance for enhancing the reasoning ability for more complicated situations. However, none of the aforementioned models can take full advantage of the semantics of relation paths.

## 3 The intuitions

The semantics of relation path is a semantic interpretation via composition of the meanings of its components. The impact of the semantics can be observed from TransE comprising multi-step relations: For the successive triples  $h \xrightarrow{r_1} t' \xrightarrow{r_2} t$  and a single triple  $h \xrightarrow{r_3} t$ , TransE considers  $\mathbf{h} + \mathbf{r}_1 \approx \mathbf{t}'$ ,  $\mathbf{t}' + \mathbf{r}_2 \approx \mathbf{t}$  and  $\mathbf{h} + \mathbf{r}_3 \approx \mathbf{t}$ . It thus implicitly

expresses that the semantics of  $r_3$  is similar to the semantic composition of  $p = (r_1, r_2)$ . However, the semantic similarity between relation and corresponding relation paths and the useful global graph structure are ignored by previous works. As shown in Table 1, for two entity pairs (sociology, George Washington University) and (Planet of the Apes, art director) from Freebase, we provide two relations and four related relation paths (each relation is mapped to two relation paths, denoted as a and b), which are considered as having similar semantics to their respective relations. However, the semantics of relation paths cannot be captured by translation-based models, such as Trans(E, H, R), because translation-based models only exploit the direct links and do not consider the semantics of relation paths.

Based on the semantic similarity, we propose a novel relation path embedding model (RPE), whose projection and type constraints of the specific relation are extended to the specific path. Furthermore, the experimental results demonstrate that by explicitly using the additional semantics, RPE significantly outperforms all baselines on link prediction and triple classification tasks.

Figure 1 illustrates the basic idea for relation-specific and path-specific projections. Each entity is projected by  $\mathbf{M}_r$  and  $\mathbf{M}_p$  into the corresponding relation and path spaces. These different embedding spaces hold the following hypothesis: in the relation-specific space, relation  $\mathbf{r}$  is regarded as a translation from head  $\mathbf{h}_r$  to tail  $\mathbf{t}_r$ ; likewise,  $\mathbf{p}^*$ , the path representation by the composition of relation embeddings, is regarded as a translation from head  $\mathbf{h}_p$  to tail  $\mathbf{t}_p$  in the path-specific space. We design two types of compositions to dynamically construct the path-specific projection  $\mathbf{M}_p$  without extra parameters. Moreover, we also propose a novel path-specific type constraints to better distinguish similar embeddings in different latent spaces.

Another intuition is the semantics of some relation paths  $p$  is unreliable for reasoning new facts of that entity pair.

For instance, there is a common relation path  $h \xrightarrow{HasChildren} t' \xrightarrow{GraduatedFrom} t$ , but this path is meaningless for inferring additional relationships between  $h$  and  $t$ . Therefore, reliable relation paths are urgently needed. As Lao et al. suggests [17], relation paths that end in many possible tail entities are more likely to be unreliable for the entity pair. Therefore, we also rank the relation paths to select reliable relation paths. Precisely, we denote all entities constitute the entity set  $\zeta$ , and all relations constitute the relation set  $R$ . For a triple  $(h, r, t)$ ,  $P_{all} = \{p_1, p_2, \dots, p_k\}$  is the path set for the entity pair  $(h, t)$ .  $P(t|h, p_i)$ , the probability of reaching  $t$  from  $h$  following the sequence of relations indicated by  $p_i$ , can be recursively defined as follows:

If  $p_i$  is an empty path,

$$P(t|h, p_i) = \begin{cases} 1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

If  $p_i$  is not an empty path, then  $p'_i$  is defined as  $r_1, \dots, r_{m-1}$ ; subsequently,

$$P(t|h, p_i) = \sum_{t' \in End(p'_i)} P(t'|h, p'_i) \cdot P(t|t', r_m) \quad (2)$$

$End(p'_i)$  is the set of ending nodes of  $p'_i$ . RPE obtains the reliable relation paths set  $P_{filter} = \{p_1, p_2, \dots, p_z\}$  by selecting relation paths whose probabilities are above a certain threshold  $\eta$ .

## 4 Relation path embedding

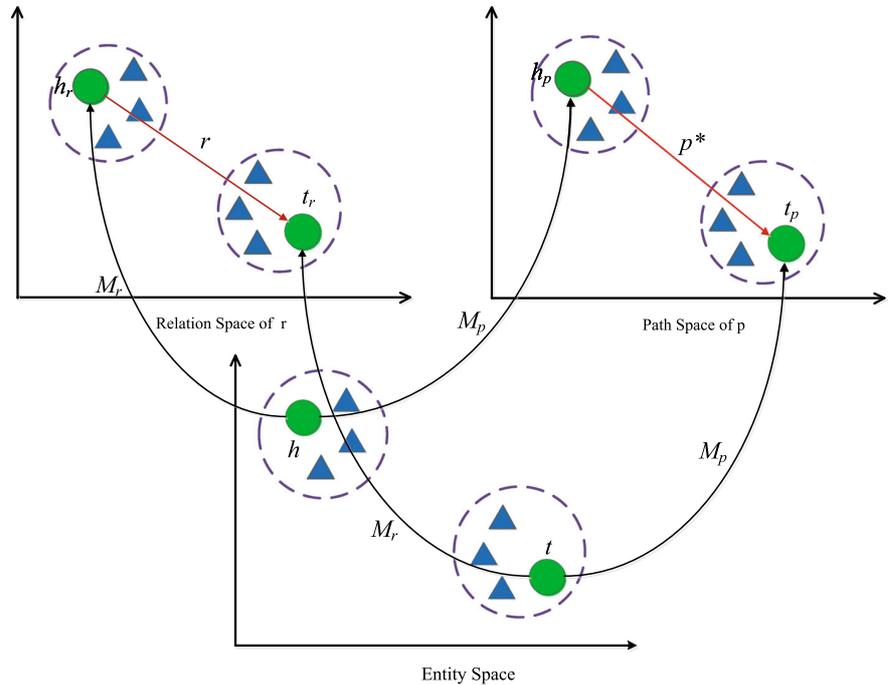
In this section, we would introduce the novel path-specific projection and type constraints and provide the details of training RPE.

**Table 1** Two examples of relation paths

entity pair	(sociology, George Washington University)
relation	/education/field_of_study/students_majoring./education/education/institution
relation paths	a: /education/field_of_study/students_majoring./education/education/student → /people/person/education./education/education/institution b: /people/person/education./education/education/major_field_of_study <sup>-1</sup> → /education/educational_institution/students_graduates./education/education/student <sup>-1</sup>
entity pair	(Planet of the Apes, art director)
relation	/education/field_of_study/students_majoring./education/education/institution
relation paths	a: /film/film/sequel → /film/film_job/films_with_this_crew_job./film/film_crew_gig/film <sup>-1</sup> b: /film/film/prequel <sup>-1</sup> → /film/film/other_crew./film/film_crew_gig/film_crew_role

$r^{-1}$  denotes the reverse of relation  $r$

**Fig. 1** Simple illustration of relation-specific and path-specific projections



### 4.1 Path-specific projection

The novel idea of RPE is that the semantics of reliable relation paths is similar to the semantics of the relation between an entity pair and both of them can be exploited for reasoning. For a triple  $(h, r, t)$ , RPE exploits projection matrices  $\mathbf{M}_r, \mathbf{M}_p \in \mathbb{R}^{m \times n}$  to project entity vectors  $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n$  in entity space into the corresponding relation and path spaces simultaneously ( $m$  is the dimension of relation embeddings,  $n$  is the dimension of entity embeddings, and  $m$  may differ from  $n$ ). The projected vectors  $(\mathbf{h}_r, \mathbf{h}_p, \mathbf{t}_r, \mathbf{t}_p)$  in their respective embedding spaces are denoted as follows:

$$\mathbf{h}_r = \mathbf{M}_r \mathbf{h}, \quad \mathbf{h}_p = \mathbf{M}_p \mathbf{h} \tag{3}$$

$$\mathbf{t}_r = \mathbf{M}_r \mathbf{t}, \quad \mathbf{t}_p = \mathbf{M}_p \mathbf{t} \tag{4}$$

Because relation paths are those sequences of relations  $p = (r_1, r_2, \dots, r_m)$ , we dynamically use  $\mathbf{M}_r$  to construct  $\mathbf{M}_p$  to decrease the model complexity. Subsequently, we explore two compositions for the formation of  $\mathbf{M}_p$ , which are formulated as follows:

$$\mathbf{M}_p = \mathbf{M}_{r_1} + \mathbf{M}_{r_2} + \dots + \mathbf{M}_{r_m} \tag{5}$$

*(addition composition)*

$$\mathbf{M}_p = \mathbf{M}_{r_1} \times \mathbf{M}_{r_2} \times \dots \times \mathbf{M}_{r_m} \tag{6}$$

*(multiplication composition)*

where addition composition (ACOM) and multiplication composition (MCOM) represent cumulative addition and multiplication for path projection. Matrix normalization is applied on  $\mathbf{M}_p$  for both compositions. The new score function is defined as follows:

$$\begin{aligned} G(h, r, t) &= S(h, r, t) + \lambda \cdot S(h, p, t) \\ &= \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\| + \frac{\lambda}{Z} \sum_{p_i \in P_{\text{filter}}} P(t|h, p_i) \cdot P_r(r|p_i) \\ &\quad \cdot \|\mathbf{h}_p + \mathbf{p}_i^* - \mathbf{t}_p\| \end{aligned} \tag{7}$$

For path representation  $\mathbf{p}^*$ , we use  $\mathbf{p}^* = \mathbf{r}_1 + \mathbf{r}_2 + \dots + \mathbf{r}_m$ , as suggested by PTransE.  $\lambda$  is the hyperparameter used to balance the knowledge embedding score  $S(h, r, t)$  and the relation path embedding score  $S(h, p, t)$ .  $Z = \sum_{p_i \in P_{\text{filter}}} P(t|h, p_i)$  is the normalization factor, and  $P_r(r|p_i) = P_r(r, p_i) / P_r(p_i)$  is used to assist in calculating the reliability of relation paths. In the experiments, we increase the limitation on these embeddings, i.e.,  $\|\mathbf{h}\|_2 \leq 1$ ,  $\|\mathbf{t}\|_2 \leq 1$ ,  $\|\mathbf{r}\|_2 \leq 1$ ,  $\|\mathbf{h}_r\|_2 \leq 1$ ,  $\|\mathbf{t}_r\|_2 \leq 1$ ,  $\|\mathbf{h}_p\|_2 \leq 1$ , and  $\|\mathbf{t}_p\|_2 \leq 1$ . By exploiting the semantics of reliable relation paths, RPE embeds entities into the

relation and path spaces simultaneously. This method considers the vital importance of reliable relation paths and improves the flexibility of modeling more complicated relations with multi-mapping properties.

### 4.2 Path-specific type constraints

In RPE, based on the semantic similarity between relations and reliable relation paths, we extend the relation-specific type constraints to novel path-specific type constraints. In type-constrained TransE, the distribution of corrupted triples is a uniform distribution.

In our model, we incorporate the two type constraints with a Bernoulli distribution. For each relation  $r$ , we denote the  $Domain_r$  and  $Range_r$  to indicate the subject and object types of relation  $r$ .  $\zeta_{Domain_r}$  is the entity set whose entities conform to  $Domain_r$ , and  $\zeta_{Range_r}$  is the entity set whose entities conform to  $Range_r$ . We calculate the average numbers of tail entities for each head entity, named  $teh$ , and the average numbers of head entities for each tail entity, named  $het$ . The Bernoulli distribution with parameter  $\frac{teh}{teh+het}$  for each relation  $r$  is incorporated with the two type constraints, which can be defined as follows: RPE samples entities from  $\zeta_{Domain_r}$  to replace the head entity with probability  $\frac{teh}{teh+het}$ , and it samples entities from  $\zeta_{Range_r}$  to replace the tail entity with probability  $\frac{het}{teh+het}$ . The objective function for RPE is defined as follows:

$$L = \sum_{(h,r,t) \in C} [L(h,r,t) + \frac{\lambda}{Z} \sum_{p_i \in P_{filter}} P(t|h,p_i) \cdot P_r(r|p_i)L(h,p_i,t)] \tag{8}$$

$L(h,r,t)$  is the loss function for triples, and  $L(h,p_i,t)$  is the loss function for relation paths.

$$L(h,r,t) = \sum_{(h',r,t') \in C''} \max(0, S(h,r,t) + \gamma_1 - S(h',r,t')) \tag{9}$$

$$L(h,p_i,t) = \sum_{(h',r,t') \in C''} \max(0, S(h,p_i,t) + \gamma_2 - S(h',p_i,t')) \tag{10}$$

We denote  $C = \{(h_i, r_i, t_i) \mid i = 1, 2, \dots, t\}$  as the set of all observed triples and

$$C' = \{(h'_i, r_i, t_i) \cup (h_i, r_i, t'_i) \mid i = 1, 2, \dots, t\}$$

as the set of corrupted triples, where each element of  $C'$  is obtained by randomly sampling from  $\zeta$ .  $C''$ , whose element conforms to the two type constraints with a Bernoulli distribution, is a subset of  $C'$ . The  $Max(0, x)$  returns the maximum between 0 and  $x$ .  $\gamma_1$  and  $\gamma_2$  are the hyperparameters of margin, which separate correct triples and corrupted triples. By exploiting these two types of prior knowledge, RPE could better distinguish similar embeddings in different embedding spaces, thus allowing it to achieve better prediction quality.

### 4.3 Training details

We adopt stochastic gradient descent (SGD) to minimize the objective function. TransE or RPE (initial) can be exploited for the initializations of all entities and relations in practice. The score function of RPE (initial) without using relation and path projections is as follows:

$$\begin{aligned} G(h,r,t) &= S(h,r,t) + \lambda \cdot S(h,p,t) \\ &= \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| + \frac{\lambda}{Z} \sum_{p_i \in P_{filter}} P(t|h,p_i) \cdot P_r(r|p_i) \\ &\quad \cdot \|\mathbf{h} + \mathbf{p}_i^* - \mathbf{t}\| \end{aligned} \tag{11}$$

We also employ this score function in our experiment as a baseline. The projection matrices  $\mathbf{M}$ s are initialized as identity matrices. RPE holds the local closed-world assumption (LCWA) [9], where each relation's domain and range types are based on the instance level. Thus, the type information is collected by accurate data provided by knowledge graph or the entities shown in observed triples.

**Algorithm 1** Learning RPE

**Input:** Training set  $C = \{(h, r, t)\}$ , entity set  $\zeta$ , relation set  $R$ , entity and relation embedding dimensions  $n$  and  $m$ , margins  $\gamma_1$  and  $\gamma_2$ , balance factor  $\lambda$ .  
**Output:** All knowledge embeddings of  $e$  and  $r$ , where  $e \in \zeta$  and  $r \in R$ .  
1: Pick up reliable relation path set  $P_{filter} = \{p\}$  for each relation by Equations 1, 2.  
2: **for all**  $k \in \zeta \cup R$  **do**  
3:    $k \in \zeta$ :  $k \leftarrow \text{Uniform}(\frac{-6}{\sqrt{n}}, \frac{-6}{\sqrt{n}})$   
4:    $k \in R$ :  $k \leftarrow \text{Uniform}(\frac{-6}{\sqrt{m}}, \frac{-6}{\sqrt{m}})$   
5: **end for**  
6: **for all**  $r \in R$  **do**  
7:   Projection matrix  $M_r \leftarrow$  identity matrix  
8: **end for**  
9: **loop**  
10:    $C_{batch} \leftarrow \text{Sample}(C, B)$  // sample a mini-batch of size  $B$   
11:    $T_{batch} \leftarrow \emptyset$  // pairs of triples for learning  
12:   **for**  $(h, r, t) \in C_{batch}$  **do**  
13:      $(h', r, t') \leftarrow$  sampling a negative triple conformed to two type constraints  
14:      $T_{batch} \leftarrow T_{batch} \cup \{(h, r, t), (h', r, t')\}$   
15:   **end for**  
16:   Update embeddings based on Equations 7, 11 and 12  
w.r.t.  $L = \sum_{(h,r,t) \in T_{batch}} [L(h, r, t) + \frac{\lambda}{Z} \sum_{p_i \in P_{filter}} P(t|h, p_i) \cdot P_r(r|p_i)L(h, p_i, t)]$ ,  
where  $L(h, r, t) = \sum_{(h', r, t') \in T_{batch}} \max(0, S(h, r, t) + \gamma_1 - S(h', r, t'))$   
and  $L(h, p_i, t) = \sum_{(h', r, t') \in T_{batch}} \max(0, S(h, p_i, t) + \gamma_2 - S(h', p_i, t'))$   
17:   Regularize the embeddings and projection matrices in  $T_{batch}$   
18: **end loop**

Note that each relation  $r$  has the reverse relation  $r^{-1}$ ; therefore, to increase supplemental path information, RPE utilizes the reverse relation paths. For example, for the relation path LeBron James  $\xrightarrow{\text{PlayFor}}$  Cavaliers  $\xrightarrow{\text{BelongTo}}$  NBA, its reverse relation path can be defined as NBA  $\xrightarrow{\text{BelongTo}^{-1}}$  Cavaliers  $\xrightarrow{\text{PlayFor}^{-1}}$  LeBron James. For each iteration, we randomly sample a correct triple  $(h, r, t)$  with its reverse  $(t, r^{-1}, h)$ , and the final score function of our model is defined as follows:

$$F(h, r, t) = G(h, r, t) + G(t, r^{-1}, h) \tag{12}$$

Theoretically, we can arbitrarily set the length of the relation path, but in the implementation, we prefer to take a smaller value to reduce the time required to enumerate all relation paths. Moreover, as suggested by the path-constrained random walk probability  $P(t|h, p)$ , as the path length increases,  $P(t|h, p)$  will become smaller and the relation path will more likely be cast off.

In our model, we particularly concentrate on how to better utilize presented relation paths to promote the performance of knowledge representation learning. Furthermore, based on the semantics of relation paths, RPE is a general framework. More sophisticated relation projection, type constraints and external information are compatible

**Table 2** Number of model parameters and computational complexity

Model	#Parameters	#Time complexity
SE [3]	$O(N_e n + 2N_r m^2)$	$O(2n^2 N_r)$
SME (linear) [4]	$O(N_e n + N_r m + 4k(n + 1))$	$O(4nkN_r)$
SME (bilinear) [4]	$O(N_e n + N_r m + 4k(ns + 1))$	$O(4nksN_r)$
TransE [5]	$O(N_e n + N_r m)$	$O(N_r)$
TransH [32]	$O(N_e n + 2N_r m)$	$O(2nN_r)$
TransR [20]	$O(N_e n + N_r(n + 1)m)$	$O(2mnN_r)$
PTransE (ADD, 2-hop) [19]	$O(N_e n + N_r m)$	$O(pN_r)$
PTransE (MUL, 2-hop) [19]	$O(N_e n + N_r m)$	$O(plN_r)$
RPE (PC) (this paper)	$O(N_e n + N_r m)$	$O(pN_r)$
RPE (ACOM) (this paper)	$O(N_e n + N_r(n + 1)m)$	$O(2mnpN_r)$
RPE (PC + ACOM) (this paper)	$O(N_e n + N_r(n + 1)m)$	$O(2mnpN_r)$

**Table 3** Statistics of the datasets

Dataset	#Ent	#Rel	#Train	#Valid	#Test
FB15K	14,591	1345	483,142	50,000	59,071
FB13	75,043	13	316,232	5908	23,733
WN11	38,696	11	112,581	2609	10,544

with it, and they can be easily incorporated into RPE to get further improvements. Algorithm 1 gives the detailed procedure of learning RPE with path-specific projection and type constraints.

#### 4.4 Complexity analysis

Table 2 lists the complexity (model parameters and time complexity in an epoch) of classical baselines and our models. We denote RPE only with path constraints as RPE (PC), RPE only with ACOM path projection as RPE (ACOM).  $N_e$  and  $N_r$  represent the numbers of entities and relations.  $N_t$  denotes the number of triples for training.  $p$  is the expected number of relation paths between the entity pair, and  $l$  is the expected length of relation paths.  $n$  is the dimension of entity embedding and  $m$  is the dimension of relation embedding.  $k$  is the number of hidden units of a neural network and  $s$  is the number of slice of a tensor. It is noticed that RPE (PC) has the same number of model parameters of TransE  $O(N_e n + N_r m)$  and RPE (ACOM), RPE (PC + ACOM) have the same numbers of model parameters of TransR  $O(N_e n + N_r (n + 1)m)$ . Both TransE and TransR are the classical models. In terms of time complexity, RPE (PC) has the same magnitude of PTransE(ADD, 2-hop)  $O(pN_t)$ , both RPE(ACOM) and RPE(PC + ACOM) retain in a magnitude  $O(2mnpN_t)$  similar to TransR  $O(2mnN_t)$ . In practice, our models take roughly 9 h for training, which can be accelerated by increasing the value of threshold  $\eta$ . For comparison purposes, the efficient implementations of classical knowledge embedding models<sup>1</sup> are exploited.

## 5 Experiments

We evaluate our model on two classical large-scale knowledge graphs: Freebase and WordNet. Freebase is a large collaborative knowledge graph that contains billions of facts about the real world, such as the triple (Beijing, Locatedin, China), which describes the fact that Beijing is located in China. WordNet is a large lexical knowledge base of English, in which each entity is a synset that

expresses a distinct concept, and each relationship is a conceptual-semantic or lexical relation. We use two subsets of Freebase, FB15K and FB13 [5], and one subset of WordNet, WN11 [27]. These datasets are commonly used in various methods, and Table 3 presents the statistics of them, where each column represents the numbers of entity type, relation type and triples that have been split into training, validation and test sets.

In our model, each triple has its own reverse triple for increasing the reverse relation paths. Therefore, the total number of triples is twice as many as the original datasets. We utilize the type information provided by [34] for FB15K. As for FB13 and WN11, we exploit the LCWA to collect the type information, so we do not depend on the auxiliary data, the domain and range of each relation are approximated by the observed triples.

To examine the retrieval and discriminative ability of our model, RPE is evaluated on two standard subtasks of knowledge graph completion: link prediction and triple classification. The experimental results and analyses are concluded in the following two subsections.

### 5.1 Link prediction

The link prediction is a classical evaluation of knowledge graph completion, it focuses on predicting the possible  $h$  or  $t$  for test triples when  $h$  or  $t$  is missed. Rather than requiring one best answer, it places more emphasis on ranking a list of candidate entities from the knowledge graph. FB15K is benchmark dataset to be employed for this task.

#### 5.1.1 Evaluation protocol

We follow the same evaluation procedures as used in knowledge embedding models. First, for each test triple  $(h, r, t)$ , we replace  $h$  or  $t$  with every entity in  $\zeta$ . Second, each corrupted triple is calculated by the corresponding score function  $S(h, r, t)$ . The final step is to rank the original correct entity with these scores in descending order.

Two evaluation metrics are reported: the average rank of correct entities (Mean Rank) and the proportion of correct entities ranked in the top 10 (Hits@10). Note that if a corrupted triple already exists in the knowledge graph, then it should not be considered to be incorrect. We prefer to remove these corrupted triples from our dataset and call this setting “filter”. If these corrupted triples are reserved, then we call this setting “raw”. In both settings, if the latent representations of entity and relation are better, then a lower mean rank and a higher Hits@10 should be achieved. Because we use the same dataset, the baseline results reported in [5, 14, 15, 19, 20, 32] are used for comparison.

<sup>1</sup> <https://github.com/thunlp/Fast-TransX>.

**Table 4** Evaluation results on link prediction

Metric	Mean rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
E [3]	273	162	28.8	39.8
LFM [13]	283	164	26.0	33.1
TransE [5]	243	125	34.9	47.1
SME (linear) [4]	274	154	30.7	40.8
SME (bilinear) [4]	284	158	31.3	41.3
TransH (unif) [32]	211	84	42.5	58.5
TransH (bern) [32]	212	87	45.7	64.4
TransR (unif) [20]	226	78	43.8	65.5
TransR (bern) [20]	198	77	48.2	68.7
TransD (unif) [14]	211	67	49.4	74.2
TransD (bern) [14]	194	91	<b>53.4</b>	77.3
TranSparse (separate, S, unif) [15]	211	63	50.1	77.9
TranSparse (separate, S, bern) [15]	187	82	53.3	79.5
PTransE (ADD, 2-hop) [19]	200	54	51.8	83.4
PTransE (MUL, 2-hop) [19]	216	67	47.4	77.7
PTransE (ADD, 3-hop) [19]	207	58	51.4	84.6
DKRL(CNN)+TransE [33]	181	91	49.6	67.4
RPE (initial)	207	58	50.8	82.2
RPE (PC)	196	77	49.1	72.6
RPE (ACOM)	<b>171</b>	<b>41</b>	52.0	<b>85.5</b>
RPE (MCOM)	183	43	52.2	81.7
RPE (PC + ACOM)	184	42	51.1	84.2
RPE (PC + MCOM)	186	43	51.7	76.5

### 5.1.2 Implementation

We set the dimension of entity embedding  $n$  and relation embedding  $m$  among  $\{20, 50, 100, 120\}$ , the margin  $\gamma_1$  among  $\{1, 2, 3, 4, 5\}$ , the margin  $\gamma_2$  among  $\{3, 4, 5, 6, 7, 8\}$ , the learning rate  $\alpha$  for SGD among  $\{0.01, 0.005, 0.0025, 0.001, 0.0001\}$ , the batch size  $B$  among  $\{20, 120, 480, 960, 1440, 4800\}$ , and the balance factor  $\lambda$  among  $\{0.5, 0.8, 1, 1.5, 2\}$ . The threshold  $\eta$  was set in the range of  $\{0.01, 0.02, 0.04, 0.05\}$  to reduce the calculation of meaningless paths.

Grid search is used to determine the optimal parameters. The best configurations for RPE (ACOM) are  $n = 100$ ,  $m = 100$ ,  $\gamma_1 = 2$ ,  $\gamma_2 = 5$ ,  $\alpha = 0.0001$ ,  $B = 4800$ ,  $\lambda = 1$ , and  $\eta = 0.05$ . We select RPE (initial) to initialize our model, set the path length as 2, take  $L_1$  norm for the score function, and traverse our model for 500 epochs.

### 5.1.3 Analysis of results

Table 4 reports the results of link prediction, in which the first column is translation-based models, the variants of

PTransE, Multi-source information learning model DKRL, and our models. As mentioned above, RPE only with MCOM path projection is denoted as RPE (MCOM). The numbers in bold are the best performance, and  $n$ -hop indicates the path length  $n$  that PTransE exploits. From the results, we can observe the followings.

(1) Our models significantly outperform the classical knowledge embedding models (TransE, TransH, TransR, TransD, and TranSparse) and PTransE on FB15K with the metrics of mean rank and Hits@10 (“filter”). Most notably, compared with TransE, mean rank reduces by 29.6, 67.2% and Hits@10 rises by 49.0 and 81.5% in RPE (ACOM). The results demonstrate that the path-specific projection can make better use of the semantics of relation paths, which are crucial for knowledge graph completion. (2) RPE with path-specific type constraints and projection (RPE (PC + ACOM) and RPE (PC + MCOM)) is a compromise between RPE (PC) and RPE (ACOM). RPE (PC) improves slightly compared with the baselines. We believe that this result is primarily because RPE (PC) only focuses on local information provided by related embeddings, ignoring some global information compared with the approach of randomly selecting corrupted entities. (3) In terms of mean rank, RPE (ACOM) has shown the best performances. Especially compared with PTransE, it achieves 14.5 and 24.1% error reductions in the “raw” and “filter” settings, respectively. In terms of Hits@10, although RPE (ACOM) is 2.7% lower than TransD (bern) on “raw”, however, it gets 10.6% higher than the latter on “filter”.

Table 5 presents the separated evaluation results by mapping properties of relations on FB15K. Mapping properties of relations follows the same rules in [5], and the metrics are Hits@10 on head and tail entities. From Table 4, we can conclude that (1) RPE (ACOM) outperforms all baselines in all mapping properties of relations. In particular, for the 1-to-N, N-to-1, and N-to-N types of relations that plague knowledge embedding models, RPE (ACOM) improves 4.1, 4.6, and 4.9% on head entity’s prediction and 6.9, 7.0, and 5.1% on tail entity’s prediction compared with previous state-of-the-art performances achieved by PTransE (ADD, 2-hop). (2) RPE (MCOM) does not perform as well as RPE (ACOM), and we believe that this result is because RPE’s path representation is not consistent with RPE (MCOM)’s composition of projections. Although RPE (PC) improves little compared with PTransE, we will indicate the effectiveness of relation-specific and path-specific type constraints in triple classification. (3) We use the relation-specific projection to construct path-specific ones dynamically; then, entities are encoded into relation-specific and path-specific spaces simultaneously. The experiments are similar to link

**Table 5** Evaluation results on FB15K by mapping properties of relations (%)

Tasks Relation category	Predicting head entities (Hits@10)				Predicting tail entities (Hits@10)			
	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 [3]	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3
SME (linear) [4]	35.1	53.7	19.0	40.3	32.7	14.9	61.6	43.3
SME (bilinear) [4]	30.9	69.6	19.9	38.6	28.2	13.1	76.0	41.8
TransE [5]	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif) [32]	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern) [32]	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif) [20]	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern) [20]	78.8	89.2	34.1	69.2	79.2	37.4	90.4	72.1
TransD (unif) [14]	80.7	85.8	47.1	75.6	80.0	54.5	80.7	77.9
TransD (bern) [14]	86.1	95.5	39.8	78.5	85.4	50.6	94.4	81.2
PTransE (ADD, 2-hop) [19]	91.0	92.8	60.9	83.8	91.2	74.0	88.9	86.4
PTransE (MUL, 2-hop) [19]	89.0	86.8	57.6	79.8	87.8	71.4	72.2	80.4
PTransE (ADD, 3-hop) [19]	90.1	92.0	58.7	86.1	90.7	70.7	87.5	88.7
TransSparse (separate, S, unif) [15]	82.3	85.2	51.3	79.6	82.3	59.8	84.9	82.1
TransSparse (separate, S, bern) [15]	86.8	95.5	44.3	80.9	86.6	56.6	94.4	83.3
RPE (initial)	83.9	93.6	60.1	78.2	82.2	66.8	92.2	80.6
RPE (PC)	82.6	92.7	44.0	71.2	82.6	64.6	81.2	75.8
RPE (ACOM)	<b>92.5</b>	<b>96.6</b>	<b>63.7</b>	<b>87.9</b>	<b>92.5</b>	<b>79.1</b>	<b>95.1</b>	<b>90.8</b>
RPE (MCOM)	91.2	95.8	55.4	87.2	91.2	66.3	94.2	89.9
RPE (PC + ACOM)	89.5	94.3	63.2	84.2	89.1	77.0	89.7	87.6
RPE (PC + MCOM)	89.3	95.6	45.2	84.2	89.7	62.8	94.1	87.7

**Table 6** Evaluation results of triple classification (%)

Datasets	WN11	FB13	FB15K
TransE (unif) [5]	75.9	70.9	77.8
TransE (bern) [5]	75.9	81.5	85.3
TransH (unif) [32]	77.7	76.5	78.4
TransH (bern) [32]	78.8	83.3	85.8
TransR (unif) [20]	85.5	74.7	79.2
TransR (bern) [20]	85.9	82.5	87.0
PTransE (ADD, 2-hop) [19]	80.9	73.5	83.4
PTransE (MUL, 2-hop) [19]	79.4	73.6	79.3
PTransE (ADD, 3-hop) [19]	80.7	73.3	82.9
RPE (initial)	80.2	73.0	68.8
RPE (PC)	83.8	77.4	77.9
RPE (ACOM)	84.7	80.9	85.4
RPE (MCOM)	83.6	76.2	85.1
RPE (PC + ACOM)	<b>86.8</b>	<b>84.3</b>	<b>89.8</b>
RPE (PC + MCOM)	85.7	83.0	87.5

prediction, and the results of experiments further demonstrate the better expressibility of our model.

## 5.2 Triple classification

Triple classification is recently proposed by [26], it aims to predict whether a given triple  $(h,r,t)$  is true, which is a binary classification problem. We conduct this task on three benchmark datasets: FB15K, FB13 and WN11.

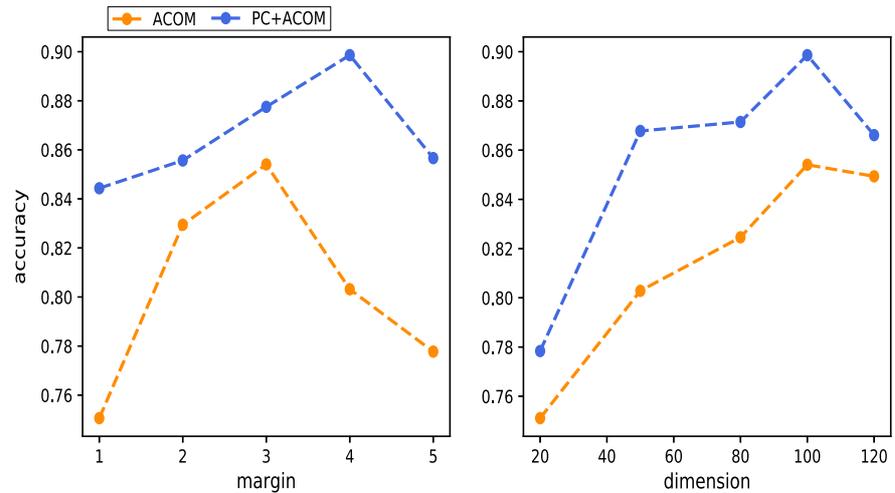
### 5.2.1 Evaluation protocol

We set different relation-specific thresholds  $\{\delta_r\}$  to perform this task. For a test triple  $(h,r,t)$ , if its score  $S(h,r,t)$  is below  $\delta_r$ , then we predict it as a positive one; otherwise, it is negative.  $\{\delta_r\}$  is obtained by maximizing the classification accuracies on the valid set.

### 5.2.2 Implementation

We directly compare our model with prior work using the results about knowledge embedding models reported in [20] for WN11 and FB13. Because [19] does not evaluate PTransE's performance on this task, we use the code of PTransE that is released in [19] to complete it. FB13 and WN11 already contain negative samples. For FB15K, we use the same process to produce negative samples, as

**Fig. 2** Hyperparameter tuning for RPE. The vertical axis is the accuracy of triple classification. The left horizontal axis represents the margin  $\gamma_1$  and the right represents the dimension of entity embedding  $n$



suggested by [27]. The hyperparameter intervals are the same as link prediction. The best configurations for RPE (PC + ACOM) are as follows:  $n = 50$ ,  $m = 50$ ,  $\gamma_1 = 5$ ,  $\gamma_2 = 6$ ,  $\alpha = 0.0001$ ,  $B = 1440$ ,  $\lambda = 0.8$ , and  $\eta = 0.05$ , taking the  $L_1$  norm on WN11;  $n = 100$ ,  $m = 100$ ,  $\gamma_1 = 3$ ,  $\gamma_2 = 6$ ,  $\alpha = 0.0001$ ,  $B = 960$ ,  $\lambda = 0.8$ , and  $\eta = 0.05$ , taking the  $L_1$  norm on FB13; and  $n = 100$ ,  $m = 100$ ,  $\gamma_1 = 4$ ,  $\gamma_2 = 5$ ,  $\alpha = 0.0001$ ,  $B = 4800$ ,  $\lambda = 1$ , and  $\eta = 0.05$ , taking the  $L_1$  norm on FB15K. We exploit RPE (initial) for initiation, and we set the path length as 2 and the maximum epoch as 500.

### 5.2.3 Analysis of results

Table 6 lists the results for triple classification on different datasets, and the evaluation metric is classification accuracy. The results demonstrate that (1) RPE (PC + ACOM), which takes good advantage of path-specific projection and type constraints, achieves the best performance on all datasets; (2) RPE (PC) achieves 4.5, 6.0 and 13.2% improvements on different datasets compared with RPE (initial); thus, path-specific constraints can raise the determinability of knowledge embedding model experimentally. Moreover, lengthening the distances for similar entities in embedding space is essential to specific tasks. (3) In contrast to WN11 and FB13, the significant improvements on FB15K indicate that although LCWA can compensate for the loss for type information, the actual relation-type information is predominant.

### 5.2.4 Hyperparameter selection

Theoretically speaking, our models are most affected by learning rate, margin and embedding dimension. RPE exploits the pre-trained models for training. Thus we choose a relatively small learning rate for refinement, such as 0.0001.

We make a comparison of RPE (ACOM) and RPE (PC + ACOM) with different hyperparameters (margin  $\gamma_1$  and dimension of entity embedding  $n$ ) as shown in Figure 2.  $\gamma_1$  is tuned in the definite range  $\{1, 2, 3, 4, 5\}$  and  $n$  is tuned in the definite range  $\{20, 40, 60, 80, 100, 120\}$ . Other parameters keep the optimal settings explained in Sect. 5.2.2. From the figure, we can see that: (1) Even without hyperparameter selection RPE can maintain high classification ability. The whole accuracies of RPE (ACOM) and RPE (PC + ACOM) exceed 75 and 77%, respectively. (2) In the condition of fine granularity, the margin and embedding dimension play an important role in model learning.

## 6 Conclusions and future work

In this paper, we propose a novel relation path embedding model (RPE) for knowledge graph completion. To the best of our knowledge, this is the first time that a path-specific projection has been proposed, and it simultaneously embeds each entity into relation and path spaces to learn more meaningful embeddings. We also put forward the novel path-specific type constraints based on relation-specific constraints to better distinguish similar embeddings in the latent space. Extensive experiments show that RPE outperforms all baseline models on link prediction and triple classification tasks. Moreover, the path-specific projection and constraints introduced by RPE are quite open strategies which can be easily incorporated into many other translation-based and multi-source information learning models.

In the future, we will explore the following directions: (1) Incorporate other potential semantic information into the relation path modeling, such as the information provided by those intermediate nodes connected by relation paths. (2) With the booming of deep learning for natural language processing [35], we plan to use convolutional

neural networks or recurrent neural networks to learn relation path representation in an end-to-end fashion. (3) Explore relation path embedding in other applications associated with knowledge graphs, such as distant supervision for relation extraction and question answering over knowledge graphs.

**Acknowledgements** The authors are grateful for the support of the National Natural Science Foundation of China (Nos. 61572228, 61472158, 61300147, and 61602207), the Science Technology Development Project from Jilin Province (No. 20160101247JC), the Zhuhai Premier Discipline Enhancement Scheme, the Guangdong Premier Key-Discipline Enhancement Scheme, and the Smart Society Collaborative Project funded by the European Commission's 7th Framework ICT Programme for Research and Technological Development under Grant Agreement No. 600854.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Ahn S, Choi H, Pärnamaa T, Bengio Y (2016) A neural knowledge language model. [arXiv:1608.00318](https://arxiv.org/abs/1608.00318)
- Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of KDD, pp 1247–1250
- Bordes A, Weston J, Collobert R, Bengio Y (2011) Learning structured embeddings of knowledge bases. In: Proceedings of AAAI, pp 301–306
- Bordes A, Glorot X, Weston J, Bengio Y (2013) A semantic matching energy function for learning with multi-relational data. *Mach Learn* 94:233–259
- Bordes A, Usunier N, García-Durán A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS, pp 2787–2795
- Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka E, Mitchell T (2010) Toward an architecture for never-ending language learning. In: Proceedings of AAAI, pp 1306–1313
- Chang KW, Tau YW, Yang B, Meek C (2014) Typed tensor decomposition of knowledge bases for relation extraction. In: Proceedings of EMNLP, pp 1568–1579
- Dong L, Wei F, Zhou M, Xu K (2015) Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of ACL, pp 260–269
- Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of KDD, pp 601–610
- García-Durán A, Bordes A, Usunier N (2015) Composing relationships with translations. In: Proceedings of EMNLP, pp 286–290
- Gardner M, Mitchell T (2015) Efficient and expressive knowledge base completion using subgraph feature extraction. In: Proceedings of EMNLP, pp 1488–1498
- Guu K, Miller J, Liang P (2015) Traversing knowledge graphs in vector space. In: Proceedings of EMNLP, pp 318–327
- Jenatton R, Roux NL, Bordes A, Obozinski G (2012) A latent factor model for highly multi-relational data. In: Proceedings of NIPS, pp 3167–3175
- Ji G, He S, Xu L, Liu K, Zhao J (2015) Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of ACL, pp 687–696
- Ji G, Liu K, He S, Zhao J (2016) Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of AAAI, pp 985–991
- Krompass D, Baier S, Tresp V (2015) Type-constrained representation learning in knowledge graphs. In: Proceedings of ISWC, pp 640–655
- Lao N, Mitchell T, Cohen WW (2011) Random walk inference and learning in a large scale knowledge base. In: Proceedings of EMNLP, pp 529–539
- Lao N, Minkov E, Cohen WW (2015) Learning relational features with backward random walks. In: Proceedings of ACL, pp 666–675
- Lin Y, Liu Z, Luan H, Sun M, Rao S, Liu S (2015) Modeling relation paths for representation learning of knowledge bases. In: Proceedings of EMNLP, pp 705–714
- Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015) Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp 2181–2187
- Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS, pp 3111–3119
- Miller GA (1995) Wordnet: a lexical database for english. *Commun ACM* 38:39–41
- Neelakantan A, Roth B, McCallum A (2015) Compositional vector space models for knowledge base completion. In: Proceedings of ACL, pp 156–166
- Nickel M, Murphy K, Tresp V, Gabrilovich E (2016) A review of relational machine learning for knowledge graphs. *Proc IEEE* 104:11–33
- Richardson M, Domingos PM (2006) Markov logic networks. *Mach Learn* 62:107–136
- Riedel S, Yao L, McCallum A, Marlin BM (2013) Relation extraction with matrix factorization and universal schemas. In: Proceedings of HLT-NAACL, pp 74–84
- Socher R, Chen D, Manning CD, Ng AY (2013) Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of NIPS, pp 926–934
- Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: Proceedings of WWW, pp 697–706
- Toutanova K, Lin V, tau Yih W, Poon H, Quirk C (2016) Compositional learning of embeddings for relation paths in knowledge base and text. In: Proceedings of ACL, pp 1434–1444
- Wang Q, Wang B, Guo L (2015) Knowledge base completion using embeddings and rules. In: Proceedings of IJCAI, pp 1859–1865
- Wang Q, Liu J, Luo Y, Wang B, Lin CY (2016) Knowledge base completion via coupled path ranking. In: Proceedings of ACL, pp 1308–1318
- Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp 1112–1119
- Xie R, Liu Z, Jia J, Luan H, Sun M (2016) Representation learning of knowledge graphs with entity descriptions. In: Proceedings of AAAI, pp 2659–2665
- Xie R, Liu Z, Sun M (2016) Representation learning of knowledge graphs with hierarchical types. In: Proceedings of IJCAI, pp 2965–2971
- Zhang H, Li J, Ji Y, Yue H (2017) Understanding subtitles by character-level sequence-to-sequence learning. *IEEE Trans Inf Inform* 13:616–624