Representation Learning with Ordered Relation Paths
for Knowledge Graph Completion

Yao Zhu[1], Hongzhi Liu[2*], Zhonghai Wu[3,4], Yang Song[5] and **Tao Zhang[5]**

[1] Center for Data Science, Peking University, Beijing, China
[2] School of Software and Microelectronics, Peking University, Beijing, China
[3] National Engineering Center of Software Engineering, Peking University, Beijing, China
[4] Key Lab of High Confidence Software Technologies (MOE), Peking University, Beijing, China
[5] BossZhipin NLP Center @Kanzhun.com, Beijing, China

{yao.zhu,liuhz,wuzh}@pku.edu.cn, {songyang,kylen.zhang}@kanzhun.com

Abstract

Incompleteness is a common problem for existing knowledge graphs (KGs), and the completion of KG which aims to predict links between entities is challenging. Most existing KG completion methods only consider the direct relation between nodes and ignore the relation paths which contain useful information for link prediction. Recently, a few methods take relation paths into consideration but pay less attention to the order of relations in paths which is important for reasoning. In addition, these path-based models always ignore nonlinear contributions of path features for link prediction. To solve these problems, we propose a novel KG completion method named OPTransE. Instead of embedding both entities of a relation into the same latent space as in previous methods, we project the head entity and the tail entity of each relation into different spaces to guarantee the order of relations in the path. Meanwhile, we adopt a pooling strategy to extract nonlinear and complex features of different paths to further improve the performance of link prediction. Experimental results on two benchmark datasets show that the proposed model OPTransE performs better than state-of-the-art methods.

1 Introduction

Knowledge graphs (KGs) are built to store structured facts which are encoded as triples, e.g., (**Beijing**, *CapitalOf*, **China**) (Lehmann et al., 2015). Each triple (h, r, t) consists of two entities h, t and a relation r, indicating there is a relation r between h and t. Large-scale KGs such as YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995) contain billions of triples and have been widely applied in various fields (Riedel et al., 2013; Dong et al., 2015). However, a common problem with these KGs is that they are far from complete, which has limited the development of KG's applications. Thus, KG completion with the goal of filling in missing parts of the KG has become an urgent issue. Specifically, KG completion aims to predict whether a relationship between two entities is likely to be true, which is defined as the link prediction in KGs.

Most existing KG completion methods are based on representation learning, which embed both entities and relations into continuous low-dimension spaces. TransE (Bordes et al., 2013) is one of the most classical KG completion models, which embeds entities and relations into the same latent space. To better deal with complex relations like 1-to-N, N-to-1 and N-to-N, TransH (Wang et al., 2014) and TransR (Lin et al., 2015b) employ relation-specific hyperplanes and relation-specific spaces respectively to separate triples according their corresponding relation. Unfortunately, these models ignore the relation paths between entities which are helpful for reasoning. For example, if we know A is B's brother, and B is C's parent, then we can infer that A is C's uncle.

Recently, a few researchers take relation paths in KGs as additional information for representation learning and attempt to project paths into latent spaces, which get better performance compared with conventional methods. PTransE-ADD (Lin et al., 2015a) considers relation paths as translations between entities and represents each path as the vector sum of all the relations in the path. Moreover, RPE (Lin et al., 2018) extends the TransR model by incorporating the path-specific projection. However, these methods pay less attention to the order of relations in paths which is important for link prediction. Figure 1 shows an example of the meaning change when the order of relations is altered. In addition, these path-based models assume information from different paths between an entity pair only contributes to the re-
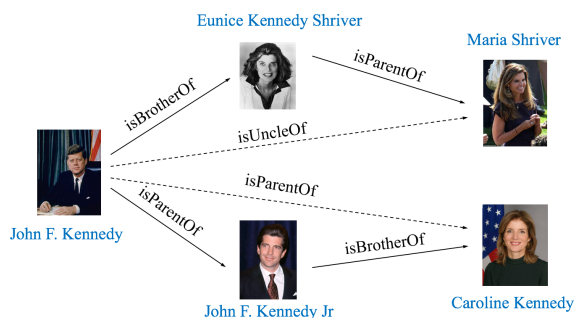
---
*Corresponding author

Figure 1: Example of the meaning change when the order of relations is altered.

lation inference linearly and ignore other complex interactions between them.

To address these issues, we propose a novel KG completion model named OPTransE. In the model, we project the head entity and the tail entity of each relation into different spaces and introduce sequence matrices to keep the order of relations in the path. Moreover, a pooling strategy is adopted to extract nonlinear features of different paths for relation inferences. Experimental results on two benchmark datasets WN18 and FB15K show that OPTransE significantly outperforms state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents the proposed model and algorithm in detail. Empirical evaluation of the proposed algorithm and comparison with other state-of-the-art algorithms are presented in Section 4. Finally, Section 5 summarises the whole paper and points out some future work.

2   Related Work

2.1   Translation-based Models

In recent years, there has been a great deal of work on representation learning for KG completion, and most studies concentrate on translation-based models. This kind of models propose to embed both entities and relations into a continuous low-dimensional vector space according to some distance-based scoring functions.

TransE (Bordes et al., 2013) is one of the most fundamental and representative translation-based models. For the entities and relations in KGs, TransE encodes them as vectors in the same space. For each fact (h, r, t), TransE believes that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. Thus, the scoring function

is defined as

$$f_r(h, t) = \| \mathbf{h} + \mathbf{r} - \mathbf{t} \|_{1/2} . \qquad (1)$$

where $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ represent the vectors of head entity h, relation r and tail entity t, respectively. If the fact (h, r, t) is true, its score $f_r(h, t)$ tends to be close to zero.

TransE is a simple and efficient method for KG completion. However, its simple structure has flaws in dealing with complicated relations like 1-to-N, N-to-1 and N-to-N. In order to address this problem, TransH (Wang et al., 2014) introduces relation-specific hyperplanes and projects entities as vectors onto the given hyperplanes. Similar to TransH, TransR (Lin et al., 2015b) also aims to cope with complicated relations. Instead of employing the hyperplane like TransH, TransR proposes a matrix $\mathbf{W_r} \in \mathbb{R}^{m \times n}$ to project entity vectors into a relation-specific space. Moreover, STransE (Nguyen et al., 2016) extends TransR by introducing two projection matrices for the head entity and the tail entity, respectively. Therefore, the head and tail entities in a triple will be projected differently into the corresponding relation space.

2.2   Incorporating Relation Paths

The models introduced so far only exploit facts observed in KGs to conduct representation learning. In fact, there is a large amount of useful information in relation paths that can be incorporated into translation-based models to improve the performance of link prediction.

Lin et al. (2015a) proposes a path-based translation model named PTransE for KG completion. It regards relation paths as translations between entities for representation learning and utilizes a path-constraint resource allocation algorithm to evaluate the reliability of relation paths. RTransE (García-Durán et al., 2015) and TransE-COMP (Guu et al., 2015) take the sum of the vectors of all relations in a path as the representation for a relation path. For the Bilinear-COMP model (Guu et al., 2015), and the PRUNED-PATHS model (Toutanova et al., 2016), they represent each relation as a diagonal matrix, and evaluate the relation path by matrix multiplication. Most recently, PaSKoGE model (Jia et al., 2018) is proposed for KG embedding by minimizing a path-specific margin-based loss function. Moreover, RPE (Lin et al., 2018), inspired by PTransE, ex-
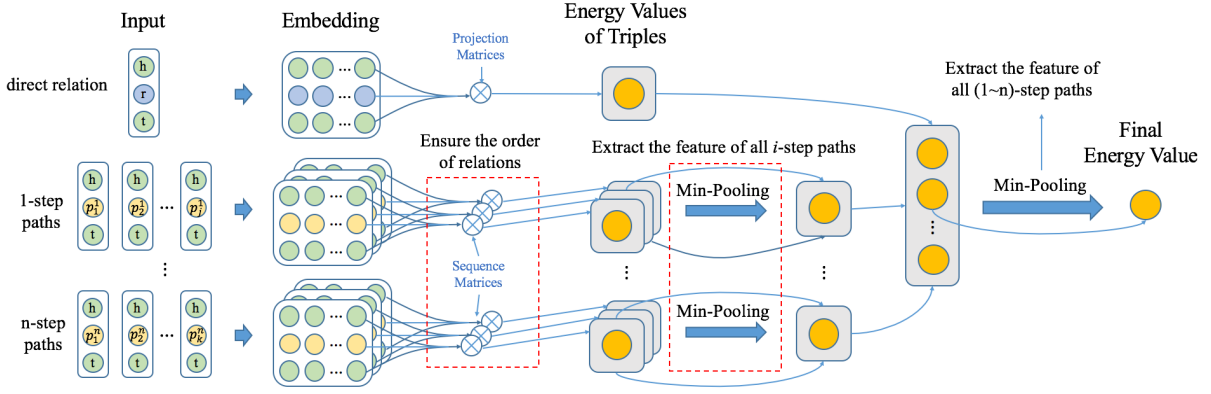
Figure 2: Architecture of OPTransE.

tends the TransR model by incorporating the path-specific projection for paths between entity pairs.

These methods try to incorporate information of relation paths to get better performance. However, they pay less attention to the order of relations in a path when learning representations of the path. In fact, changes in the relation order of paths will alter the meanings of paths to a great extent (as shown in Figure 1). Moreover, the methods stated above assume information from different paths between an entity pair only contributes to the relation inference linearly. Unfortunately, they ignore the complex nonlinear features of different paths. In order to solve these problems, we propose OPTransE, a novel KG completion model, which learns representations of ordered relation paths and designs a pooling method to better extract nonlinear features from various relation paths.

3 Our Model

To infer the missing parts of KGs, we propose a KG completion model called OPTransE, whose architecture is shown in Figure 2. We first embed the entities and relations of KG into latent spaces with the consideration of the order of relations in paths. Then, we try to infer the missing relations using these latent representations. Different from previous methods which embed the head and tail of a relation into the same latent space, we project them into different spaces. Therefore, we can distinguish the order of relations in the path. To extract the complex and nonlinear path information for relation reasoning, we design a two layer pooling strategy to fuse the information from different paths.

In this section, we will first introduce the em-

bedding representations of ordered relation paths. After that, we utilize a two layer pooling strategy to construct the total energy function of triples and then the objective function is presented. Finally, we will describe the detail of model implementation and analyze the complexity of the model.

3.1 Ordered Relation Paths Representation

For each triple (h, r, t) in KG, we employ vectors to represent the entity pair and the relation. Specifically, $\mathbf{h} \in \mathbb{R}^d$ denotes the head entity h, $\mathbf{t} \in \mathbb{R}^d$ denotes the tail entity t and $\mathbf{r} \in \mathbb{R}^d$ indicates the relation r.

We assume the paths connecting two entities contain indicative information for the direct relation between these two entities. To measure these kinds of indicative effects while guarantee the order of relations in a path, we define an energy function in Equation (2). Let $p^{s=n}$ denote one of the n-step path from h to t, i.e., $h \xrightarrow{r_1} \cdots \xrightarrow{r_n} t$. If the relation path is reasonable from h to t, it will obtain a lower energy value.

$$E\left(h, p^{s=n}, t\right) = \parallel \mathbf{h_p} + \sum_{i=1}^{n} \mathbf{S_p^i} \mathbf{r}_i - \mathbf{t_p} \parallel_{1/2}, \quad (2)$$

where

$$\mathbf{h_p} = f(p, \mathbf{h}), \quad \mathbf{t_p} = g(p, \mathbf{t}), \quad (3)$$

$$\mathbf{S_p^i} = \mathbf{W}(p, i). \quad (4)$$

$\mathbf{h_p}$ and $\mathbf{t_p}$ denote the representations of the head entity h and the tail entity t in the ordered relation path p, respectively. $\mathbf{S_p^i} \in \mathbb{R}^{d \times d}$ denotes the sequence matrix with respect to the i-th relation in the given path p.
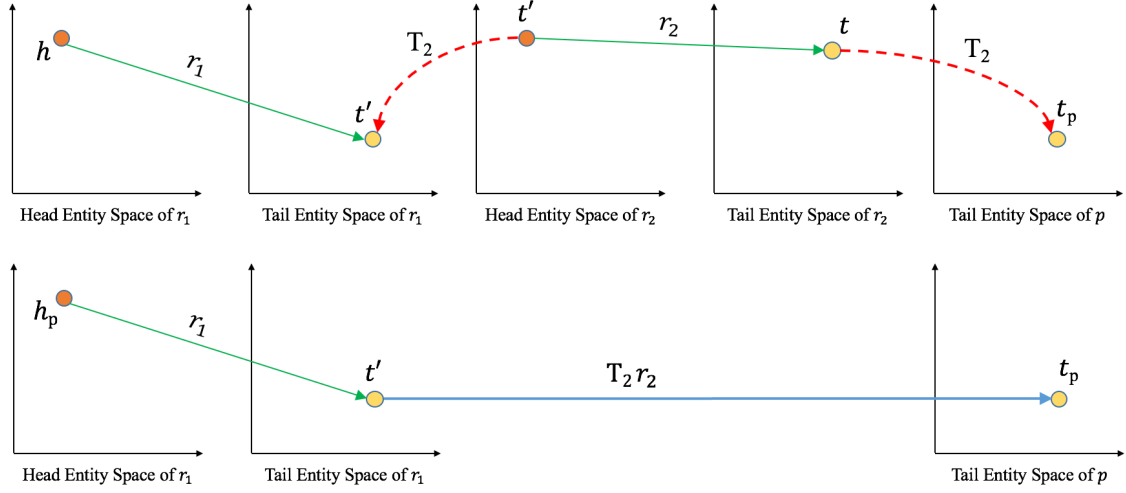
Figure 3: The representation of path p ($h \xrightarrow{r_1} t' \xrightarrow{r_2} t$). The top part of the figure depicts the process of space transition of t' and t, while the bottom part illustrates the generated continuous path from h_p to t_p after the transition.

Note that a triple (h, r, t) in the KG can be seen as a one-step path between h and t. Thus, the value of $E(h, r, t)$ is able to be obtained by substituting direct relation r as $p^{s=1}$ into Equation (2).

From Equation (2) we could observe that the sequence matrix $\mathbf{S}_{\mathbf{p}}^i$ before each relation $\mathbf{r}_i$ is different. If the order of several relations in a path is altered, the value of energy function will also change at the same time. Therefore, paths with the same relation set but different relation order will infer out distinct direct relations in our model. The specific representation of the ordered relation path will be demonstrated in the following contents.

To keep the order information of relations in paths, we project the head and tail entities of a relation into different spaces by introducing two matrices for each relation. Let $\mathbf{W}_{\mathbf{r},\mathbf{1}} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_{\mathbf{r},\mathbf{2}} \in \mathbb{R}^{d \times d}$ denote the projection matrices of the head entity and the tail entity for relation r, respectively. With these two matrices, we will project the head and tail entities into distinct spaces with respect to the same relation. Suppose there is a path $r_1, r_2, \ldots, r_n$ from h to t, ideally, we define the following equations

$$\begin{cases} \mathbf{W}_{\mathbf{r_1},\mathbf{1}}\mathbf{h} + \mathbf{r_1} = \mathbf{W}_{\mathbf{r_1},\mathbf{2}}\mathbf{t}^{(1)} \\ \mathbf{W}_{\mathbf{r_2},\mathbf{1}}\mathbf{t}^{(1)} + \mathbf{r_2} = \mathbf{W}_{\mathbf{r_2},\mathbf{2}}\mathbf{t}^{(2)} \\ \qquad\qquad \vdots \\ \mathbf{W}_{\mathbf{r_n},\mathbf{1}}\mathbf{t}^{(n-1)} + \mathbf{r_n} = \mathbf{W}_{\mathbf{r_n},\mathbf{2}}\mathbf{t} \end{cases}, \quad (5)$$

where $\mathbf{t}^{(i)}$ indicates the i-th passing node on the path.

For the entity pair with a relation path, we get their representations after eliminating the passing nodes from Equation (5). Thus, the concrete forms of the variables in Equation (2) are shown as follows,

$$\mathbf{h_p} = \mathbf{W}_{\mathbf{r_1},\mathbf{1}}\mathbf{h}, \quad \mathbf{t_p} = \mathbf{W}_{\mathbf{p}^{\mathbf{s=n}}}\mathbf{t}, \quad (6)$$

$$\mathbf{S}_{\mathbf{p}}^i = \prod_{k=1}^{i} \mathbf{T}_k, \quad (7)$$

where

$$\mathbf{W}_{\mathbf{p}^{\mathbf{s=n}}} = \mathbf{S}_{\mathbf{p}}^n \mathbf{W}_{\mathbf{r_n},\mathbf{2}}, \quad (8)$$

$$\mathbf{T}_k = \begin{cases} I & k = 1 \\ \mathbf{M}(r_k, r_{k-1}) & k > 1 \end{cases}. \quad (9)$$

$\mathbf{W}_{\mathbf{p}^{\mathbf{s=n}}} \in \mathbb{R}^{d \times d}$ indicates the projection matrix for path $p^{s=n}$, which aims to project the tail entity in a path to the space of $p^{s=n}$. Moreover, I in Equation (9) denotes the identity matrix and $\mathbf{M}(r_k, r_{k-1}) \in \mathbb{R}^{d \times d}$ means the space transition matrix from the head entity space of r_k to the tail entity space of r_{k-1}, i.e., $\mathbf{M}(r_k, r_{k-1})\mathbf{W}_{\mathbf{r_k},\mathbf{1}} = \mathbf{W}_{\mathbf{r_{k-1}},\mathbf{2}}$.

Figure 3 illustrates the representation of the relation path in our model. Suppose there is a 2-step path from h to t passing t', i.e., $h \xrightarrow{r_1} t' \xrightarrow{r_2} t$. It is obvious that t' acts as the tail entity of relation r_1 and as the head entity of relation r_2 at the same time, which is shown on the top part of Figure 3. To connect relations in different spaces, we try to

unify the passing node in the path into the same space. As defined in Equation (9), T_2 is utilized to transfer the passing node t' from the head entity space of r_2 to the tail entity space of r_1. Moreover, T_2 is also assigned to the relation r_2 and the tail entity t. Note that the tail entity t will be projected into the space of path p which is defined in Equation (6). Finally, the path from h_p to t_p will pass through r_1 and T_2r_2 as shown on the bottom part of Figure 3.

3.2 Pooling Strategy

We design a two layer pooling strategy to fuse the information from different paths. First, we utilize a minimum pooling method to extract feature information from paths with i steps and define an energy function as follows,

$$E\left(h, P_r^{s=i}, t\right) = \text{Min}[\ E\left(h, p^{s=i}, t\right) \mid p^{s=i} \in P_r^{s=i}],\tag{10}$$

where $P_r^{s=i}$ indicates the set of all i-step paths which are relevant to the relation r from the head entity h to the tail entity t. To obtain $P_r^{s=i}$, we introduce a conditional probability $\text{Pr}(r|p^{s=i})$ to represent the reliability of a path $p^{s=i}$ associated with the given relation r,

$$\begin{aligned}\text{Pr}\left(r\middle|p^{s=i}\right) &= \text{Pr}(r, p^{s=i})/\ \text{Pr}(p^{s=i})\\ &= \frac{N(r, p^{s=i})/N(p)}{N(p^{s=i})/N(p)}\\ &= \frac{N(r, p^{s=i})}{N(p^{s=i})},\end{aligned}\tag{11}$$

where $\text{Pr}(r, p^{s=i})$ denotes the joint probability of r and $p^{s=i}$, $\text{Pr}(p^{s=i})$ denotes the marginal probability of $p^{s=i}$. In addition, $N(r, p^{s=i})$ denotes the number of cases where r and $p^{s=i}$ link the same entity pair in the KG, $N(p^{s=i})$ denotes the number of the path $p^{s=i}$ in the KG and $N(p)$ denotes the total number of paths in the KG. Since $N(p)$ can be removed from both the numerator and denominator of the fractional expression, we finally convert the probability into frequency for computation.

We filter the paths by choosing all $p^{s=i}$ from h to t whose $\text{Pr}\left(r\middle|p^{s=i}\right) > 0$. Thus, $P_r^{s=i}$ is the set of all filtered $p^{s=i}$. Sometimes we could infer the fact not from the direct relation r but from the path, which means the value of $E\left(h, P_r^{s=i}, t\right)$ could possibly be less than that of $E\left(h, r, t\right)$.

Furthermore, we utilize a minimum pooling method to fuse information from paths with different lengths and define an energy function as follows,

$$\begin{aligned}E_{final}\left(h, r, t\right) = \text{Min}[\ &E\left(h, r, t\right),\ E\left(h, P_r^{s=1}, t\right),\\ &E\left(h, P_r^{s=2}, t\right), \ldots,\ E\left(h, P_r^{s=n}, t\right)],\end{aligned}\tag{12}$$

where $E\left(h, r, t\right)$ indicates the energy value of direct relation r and it is calculated by substituting r as $p^{s=1}$ into Equation (2). $E\left(h, P_r^{s=i}, t\right)$ is initialized as **infinite**, thus it will not influence the outcome of final energy function if there is no i-step path between h and t.

In summary, we adopt the min-pooling strategy twice in our model. For $E\left(h, P_r^{s=i}, t\right)$, min-pooling aims to choose the most matched path with r among all i-step paths. And for the final energy function, min-pooling tries to extract nonlinear features from paths of various lengths. In addition, the min-pooling method addresses the problem that there may be no relation paths between h and t.

3.3 Objective Function

The objective function for the proposed model OPTransE is formalized as

$$\begin{aligned}L(S) = \sum_{(h,r,t)\in S}&\left\{L\left(h, r, t\right) + \lambda \cdot \sum_i \left[\frac{1}{Z_i}\right.\right.\\ &\left.\left.\cdot \sum_{p^{s=i}\in P_r^{s=i}} \text{Pr}\left(p^{s=i}\middle|h, t\right) \cdot \text{Pr}\left(r\middle|p^{s=i}\right) \cdot L\left(h, p^{s=i}, t\right)\right]\right\},\end{aligned}\tag{13}$$

where $L\left(h, r, t\right)$ indicates the loss function for the triple (h, r, t), and $L\left(h, p^{s=i}, t\right)$ represents the loss value with respect to the relation path $p^{s=i}$. The probability $\text{Pr}\left(p^{s=i}\middle|h, t\right)$ indicates the reliability of the relation path $p^{s=i}$ given the entity pair (h, t), and $\text{Pr}(r|p^{s=i})$ denotes the reliability of a path $p^{s=i}$ associated with the given relation r. The details of $\text{Pr}\left(p^{s=i}\middle|h, t\right)$ are shown in (Lin et al., 2015a), which is computed by a path-constraint resource allocation algorithm. $Z_i = \sum_{p^{s=i}\in P_r^{s=i}} \text{Pr}\left(p^{s=i}\middle|h, t\right) \text{Pr}\left(r\middle|p^{s=i}\right)$ is a normalization factor, and λ is utilized to balance the triple loss and paths losses.

We adopt the margin-based loss in our model, i.e.,

$$L\left(h, r, t\right) = \sum_{(h',r,t')\in S'} [\gamma + E\left(h, r, t\right) - E(h', r, t')]_+,\tag{14}$$

$$L\left(h, p^{s=i}, t\right) = \sum_{(h',r,t') \in S'} [\gamma_i + E\left(h, p, t\right) - E(h', p, t')]_+,$$
(15)

where p is the simple form of $p^{s=i}$. $[x]_+ = max(x, 0)$ returns the higher one between x and 0. γ_i is the margin to separating positive and negative samples. It is noteworthy that we employ different margin γ_i for paths with different step number because the noise of energy function will be magnified as the number of steps increases. The corrupted triple set S' for (h, r, t) is denoted as follows:

$$S' = \{(h', r, t) \cup (h, r, t')\}.$$
(16)

We replace the head entity or the tail entity in the triple randomly and guarantee that the new triple is not an existing valid triple.

Our goal is to minimize the total loss. Valid relation paths will obtain lower energy value after the optimization, so that paths can sometimes replace directed relations when performing the prediction.

3.4 Parameter Learning

We utilize stochastic gradient descent (SGD) to optimize the objective function in Equation (13) and learn parameters of the model. To ensure the convergence of the model, we impose limitations to the norm of vectors, i.e., $||\mathbf{h}||_2 \leq 1$, $||\mathbf{r}||_2 \leq 1$, $||\mathbf{t}||_2 \leq 1$, $||\mathbf{W_{r,1}h}||_2 \leq 1$, $||\mathbf{W_{r,2}t}||_2 \leq 1$. Moreover, we note that the objective function defined in Equation (13) has two parts. The first part is for the basic triple and the second part is for the relation paths. To focus on the representation of ordered relation paths in the second part, we only update the parameters of relation vectors in the path when conducting the optimization of the model.

In addition, we follow PTransE (Lin et al., 2015a) to generate reverse relation r^{-1} to enlarge the training set, and the inference in KGs can be through the reverse paths. For instance, for the fact (**Honolulu**, *CapitalOf*, **Hawaii**), we will also add a fact with the reverse relation to the KG, i.e., (**Hawaii**, *CapitalOf^{-1}*, **Honolulu**).

3.5 Complexity Analysis

Let d denote the dimension of entities and relations, N_e and N_r denote the number of entities and relations, respectively. The number of model parameters for OPTransE is ($N_e d + N_r d + 2N_r d^2$), which is the same as that of STransE.

Moreover, let N_p denote the expected number of relation paths between the entity pair, N_t denote the number of triples for training, k denote the maximum length of relation paths. According to the objective function shown in Equation (13) and details of parameter learning stated in Section 3.4, the time complexity of OPTransE for optimization is $O(k^2 d^3 N_p N_t)$, which is on the same magnitude as that of RPE(MCOM) (Lin et al., 2018).

4 Experiments

4.1 Datasets

To evaluate the proposed model OPTransE, we use two benchmark datasets: WN18 and FB15K as experimental data. They are subsets of the knowledge graph WordNet (Miller, 1995) and Freebase (Bollacker et al., 2008), respectively (Bordes et al., 2013). These two datasets have been widely employed by researchers for KG completion (Jia et al., 2018; Lin et al., 2018). The statistic details of the two datasets are shown in Table 1. In our experiments, as we add triples of reverse relations to the datasets, the number of relations and training triples are doubled.

Table 1: Statistics of datasets

Dataset	#Rel	#Ent	#Train	#Valid	#Test
WN18	18	40,943	141,442	5,000	5,000
FB15K	1345	14,951	483,142	50,000	59,071

4.2 Experimental Settings

We adopt the idea from TransR (Lin et al., 2015b) and initialize the vectors and matrices of OPTransE by an existing method STransE (Nguyen et al., 2016). Following TransH (Wang et al., 2014), *Bernoulli* method is applied for generating head or tail entities when sampling corrupted triples.

As the length of paths increases, the reliability of the path will decline accordingly. To better determine the maximum length of paths for experiment, before the test on FB15K, we had evaluated OPTransE with 3-step paths on WN18. However, OPTransE (3-step) performs comparably as OPTransE (2-step) with a higher computational cost. This indicates that longer paths hardly contain more useful information and it is unnecessary

Table 2: Evaluation results on link prediction

Model	WN18				FB15K			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filtered	Raw	Filtered	Raw	Filtered	Raw	Filtered
SE	1011	985	68.5	80.5	273	162	28.8	39.8
SME	545	533	65.1	74.1	274	154	30.7	40.8
TransE	263	251	75.4	89.2	243	125	34.9	47.1
TransH	318	303	75.4	86.7	212	87	45.7	64.4
TransR	238	225	79.8	92.0	198	77	48.2	68.7
TranSparse	223	211	80.1	93.2	187	82	53.5	79.5
STransE	<u>217</u>	206	80.9	93.4	219	69	51.6	79.7
ITransF	-	<u>205</u>	-	94.2	-	65	-	81.0
HolE	-	-	-	94.9	-	-	-	73.9
ComplEx	-	-	-	94.7	-	-	-	84.0
ANALOGY	-	-	-	94.7	-	-	-	85.4
ProjE	277	260	79.4	94.9	**124**	<u>34</u>	<u>54.7</u>	<u>88.4</u>
RTransE	-	-	-	-	-	50	-	76.2
PTransE (ADD, 2-step)	235	221	<u>81.3</u>	92.7	200	54	51.8	83.4
PTransE (MUL, 2-step)	243	230	79.5	90.9	216	67	47.4	77.7
PTransE (ADD, 3-step)	238	219	81.1	94.2	207	58	51.4	84.6
PaSKoGE	-	-	<u>81.3</u>	95.0	-	-	53.1	88.0
RPE (ACOM)	-	-	-	-	171	41	52.0	85.5
RPE (MCOM)	-	-	-	-	183	43	52.2	81.7
RotatE	-	309	-	**95.9**	-	40	-	<u>88.4</u>
OPTransE	**211**	**199**	**83.2**	<u>95.7</u>	<u>136</u>	**33**	**58.0**	**89.9**

to enumerate longer paths. Therefore, considering the computational efficiency, we limit the maximum length of relation paths as 2 steps.

In our experiments, we utilize the grid search to choose the best parameters for the two datasets, respectively. The best configurations for OPTransE are as follows: the dimension of entity and relation vectors $d = 50$, the learning rate $\alpha = 0.0001$, the margin $\gamma = 5.0, \gamma_1 = 5.0, \gamma_2 = 5.5$, the balance factor $\lambda = 0.01$ on WN18; and $d = 100$, $\alpha = 0.0005$, $\gamma = 4.0$, $\gamma_1 = 4.5, \gamma_2 = 5.0$, $\lambda = 0.01$ on FB15K. In addition, L1 norm is employed for scoring and we run SGD for 2000 epochs in the training procedure.

4.3 Evaluation Metrics and Baselines

The same as in previous work (Bordes et al., 2013; Nguyen et al., 2016), we evaluate the proposed model OPTransE on the link prediction task. This task aims to predict the missing entity in a triple (h, r, t), i.e., predicting h when r and t are given, or predicting t given h and r. When testing a fact (h, r, t), we replace either head or tail entity with all entities in the dataset and calculate scores of generated triples according to Equation (12). And then we sort the entities with their scores in ascending order to locate the rank of the target entity.

For specific evaluation metrics, we employ the widely used mean rank (MR) and Hits@10 in the experiments. Mean rank indicates the average rank of correct entities and Hits@10 means the proportion of correct entities ranked in top 10. Higher Hits@10 or lower value of mean rank implies the better performance of the model on the link prediction task. Moreover, it is noted that the generated triple for test may exist in the dataset as a fact, thus such triples will affect the final rank of the target entity to some extent. Hence, we could filter out these generated triples which are facts in the dataset before ranking. If we have performed filtering, the result will be denoted as "Filtered", otherwise it will be denoted as "Raw".

Moreover, Bordes et al. (2013) defined four categories of relations in KGs by mapping their properties such as 1-to-1, 1-to-N, N-to-1 and N-to-N. Thus, experimental results of distinguishing the four different relation types have also been

Table 3: Filtered evaluation results on FB15K by mapping properties of relations(%)

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	74.6	86.6	43.7	70.6	71.5	49.0	85.0	72.9
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
TranSparse	86.8	95.5	44.3	80.9	86.6	56.6	94.4	83.3
STransE	82.8	94.2	50.4	80.1	82.4	56.9	93.4	83.1
PTransE(ADD, 2-step)	91.0	92.8	60.9	83.8	91.2	74.0	88.9	86.4
PTransE(MUL, 2-step)	89.0	86.8	57.6	79.8	87.8	71.4	72.2	80.4
PTransE(ADD, 3-step)	90.1	92.0	58.7	86.1	90.7	70.7	87.5	88.7
PaSKoGE	89.7	94.8	62.3	86.7	89.3	72.9	93.4	88.9
RPE (ACOM)	<u>92.5</u>	96.6	<u>63.7</u>	87.9	<u>92.5</u>	<u>79.1</u>	95.1	90.8
RPE (MCOM)	91.2	95.8	55.4	87.2	91.2	66.3	94.2	89.9
RotatE	92.2	<u>96.7</u>	60.2	<u>89.3</u>	92.3	71.3	<u>96.1</u>	<u>92.2</u>
OPTransE	**93.1**	**97.4**	**69.0**	**89.8**	**92.8**	**87.4**	**96.7**	**92.3**

recorded for comparison.

In the link prediction task, several competitive KG completion methods are utilized as baselines, including SE (Bordes et al., 2011), SME (Bordes et al., 2014), TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015b), TranSparse (Ji et al., 2016), STransE (Nguyen et al., 2016), ITransF (Xie et al., 2017), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016), ANALOGY (Liu et al., 2017), ProjE (Shi and Weninger, 2017), RTransE (García-Durán et al., 2015), PTransE (Lin et al., 2015a), PaSKoGE (Jia et al., 2018), RPE (Lin et al., 2018) and RotatE (Sun et al., 2019). Among them, RTransE, PTransE, PaSKoGE and RPE exploit the information of paths between entity pairs.

4.4 Results

Table 2 shows the performances of different methods on the link prediction task according to various metrics. Numbers in **bold** mean the best results among all methods and the underlined ones mean the second best. The evaluation results of baselines are from their original work, and "-" in the table means there is no reported results in prior work. Note that we implement ProjE and PTransE on WN18 using the public codes.

From Table 2 we could observe that: (1) PTransE performs better than its basic model TransE, and RPE outperforms its original method TransR. This indicates that additional informa-

tion from relation paths between entity pairs is helpful for link prediction. Note that OPTransE outperforms baselines which do not take relation paths into consideration in most cases. These results demonstrate the effectiveness of OPTransE to take advantage of the path features in the KG. (2) OPTransE performs better than previous path-based models like RTransE, PTransE, PaSKoGE and RPE on all metrics. This implies that the order of relations in paths is of great importance for reasoning, and learning representations of ordered relation paths can significantly improve the accuracy of link prediction. Moreover, the proposed pooling strategy which aims to extract nonlinear features from different relation paths also contributes to the improvements of performance.

Specific evaluation results on FB15K by mapping properties of relations (1-to-1, 1-to-N, N-to-1, and N-to-N) are shown in Table 3. Several methods which have reported these results are listed as baselines. OPTransE achieves the highest scores in all sub-tasks. We note that it is more difficult to predict head entities of N-to-1 relations and tail entities of 1-to-N relations, since the prediction accuracy on these two sub-tasks is generally lower than those of other sub-tasks. Surprisingly, OPTransE has achieved significant improvements on these two sub-tasks. Especially when predicting tail entities of 1-to-N relations, OPTransE promotes Hits@10 to 87.4% which is 8.3% higher than the best performance among

baselines. Meanwhile, since the average prediction accuracy for N-to-N relations of OPTransE on the two datasets has reached 91.1%, we can also infer that our model has strong ability to deal with N-to-N relations. OPTransE projects the head and tail entities of a triple into different relation-specific spaces, thus, it is able to better discriminate the relevant entities. Furthermore, these results also confirm that ordered relation paths between entity pairs which are exploited by OPTransE contain useful information and can help to perform more accurate inference when facing complex relations.

5 Conclusion and Future Work

In this paper, we propose a novel KG completion model named OPTransE, which aims to address the issue of relation orders in paths. In our model, we project the head entity and the tail entity of each relation into different spaces to guarantee the order of the path. In addition, a pooling method is applied to extract complex and nonlinear features from numerous relation paths. Finally, we evaluate our proposed model on two benchmark datasets and experimental results demonstrate the effectiveness of OPTransE.

In the future, we will explore the following research directions: (1) we will study the applications of the proposed models in various domains, like personalized recommendation (Liu et al., 2018); (2) we will explore other techniques to fuse the ordered relation information from different paths (Liu et al., 2019).

Acknowledgments

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD'08, pages 1247–1250. ACM.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, NIPS'13, pages 2787–2795.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 301–306.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multicolumn convolutional neural networks. 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)*, volume 1 of *ACL-IJCNLP'15*, pages 260–269.

Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2015. Composing relationships with translations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 286–290.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 318–327.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 985–991.

Yantao Jia, Yuanzhuo Wang, Xiaolong Jin, and Xueqi Cheng. 2018. Path-specific knowledge graph embedding. *Knowledge-Based Systems*, 151:37–44.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Xixun Lin, Yanchun Liang, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. 2018. Relation path embedding in knowledge graphs. *Neural Computing and Applications*, pages 1–11.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015a. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 705–714.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2181–2187.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical inference for multi-relational embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, ICML'17, pages 2168–2178. JMLR. org.

Hongzhi Liu, Yingpeng Du, and Zhonghai Wu. 2019. Aem: Attentional ensemble model for personalized classifier weight learning. *Pattern Recognition*, 96:106976.

Hongzhi Liu, Zhonghai Wu, and Xing Zhang. 2018. Cplr: Collaborative pairwise learning to rank for personalized recommendation. *Knowledge-Based Systems*, 148:31–40.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, HLT-NAACL'16, pages 460–466.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1955–1961.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, HLT-NAACL'13, pages 74–84.

Baoxu Shi and Tim Weninger. 2017. Proje: embedding projection for knowledge graph completion. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 1236–1242.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, WWW'07, pages 697–706. ACM.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. ICLR'19.

Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. 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 (Volume 1: Long Papers)*, volume 1 of *ACL'16*, pages 1434–1444.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, ICML'16, pages 2071–2080.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1112–1119.

Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. 2017. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1 of *ACL'17*, pages 950–962.