

TransRHS: A Representation Learning Method for Knowledge Graphs with Relation Hierarchical Structure

Fuxiang Zhang^{1,2}, Xin Wang^{1,2,3*}, Zhao Li¹ and Jianxin Li⁴

¹College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

²Tianjin International Engineering Institute, Tianjin University, Tianjin 300350, China

³Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300350, China

⁴School of Information Technology, Deakin University, Geelong, Victoria 3220, Australia
{zhangfx, wangx, lizh}@tju.edu.cn, jianxin.li@deakin.edu.au

Abstract

Representation learning of knowledge graphs aims to project both entities and relations as vectors in a continuous low-dimensional space. Relation Hierarchical Structure (RHS), which is constructed by a generalization relationship named *subRelationOf* between relations, can improve the overall performance of knowledge representation learning. However, most of the existing methods ignore this critical information, and a straightforward way of considering RHS may have a negative effect on the embeddings and thus reduce the model performance. In this paper, we propose a novel method named TransRHS, which is able to incorporate RHS seamlessly into the embeddings. More specifically, TransRHS encodes each relation as a vector together with a *relation-specific sphere* in the same space. Our TransRHS employs the relative positions among the vectors and spheres to model the *subRelationOf*, which embodies the inherent generalization relationships among relations. We evaluate our model on two typical tasks, i.e., link prediction and triple classification. The experimental results show that our TransRHS model significantly outperforms all baselines on both tasks, which verifies that the RHS information is significant to representation learning of knowledge graphs, and TransRHS can effectively and efficiently fuse RHS into knowledge graph embeddings.

1 Introduction

With the rapid development of Artificial Intelligence (AI), Knowledge Graphs (KGs) such as Freebase [Bollacker *et al.*, 2008], DBpedia [Auer *et al.*, 2007], and NELL [Carlson *et al.*, 2010], have become fundamental data sources that support the development of many AI-related applications. A typical KG is usually represented as multi-relational data with enormous triple facts in the form of (*head entity*, *relation*, *tail entity*), denoted as (*h*, *r*, *t*).

Even though the constructed KGs have millions of facts, they are still sparse graphs in contrast to common connec-

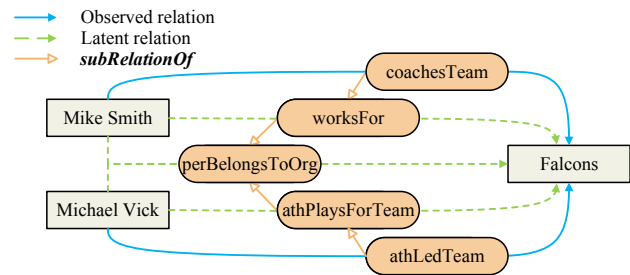


Figure 1: Example of relation hierarchical structure in NELL

tions in the real world, and thus the existing KGs need to be completed in order to improve their quality and usability. KG completion, i.e., predicting missing facts from existing ones, has become an indispensable tool to construct a high-quality KG. Recently, an effective approach for this task is representation learning, which aims to embed a knowledge graph into a continuous vector space while preserving inherent structure of the KG. Many methods have been proposed on representation learning of KGs, among which the translation-based models are considered to be most popular.

Unfortunately, the most existing translation-based representation learning methods merely focus on the structured information in triples, paying less attention to the rich information implied in RHS. [Zhang *et al.*, 2018] suggests that relations in KGs conform to a three-layer structure, i.e., semantically similar relations can construct clusters of relations and coarse-grained relations can be further split into several fine-grained sub-relations. However, the structure mentioned in that paper is actually not a real RHS, which is simply obtained by a clustering algorithm. In addition, limiting the relation structure to three layers is not universal enough considering the complexity of relations in KGs.

To address the above issue, in this paper, we introduce a more general RHS. Figure 1 shows a KG with RHS, in which (1) the rectangles represent entities, (2) the rounded rectangles represent the relations, (3) the solid lines denote the observed facts, and (4) the dashed lines denote the latent facts, respectively. Given an observed triple (Mike Smith, coachesTeam, Falcons), then the triple (Mike Smith, worksFor, Falcons) and the triple (Mike Smith,

*Xin Wang is the corresponding author.

personBelongsToOrganization, Falcons) can be logically derived according to RHS among these relations. Fusing RHS information into KG embeddings can intuitively improve the prediction ability of the model, however, in the meanwhile, it may also introduce the following influences: (1) when an existing model has been trained, a direct way of considering the RHS information will change the embeddings of relations. (2) In addition, most translation-based models satisfy $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, so the embeddings of entities will also be affected. These two issues will break the original model and reduce the model performance. In order to make full use of RHS while eliminating this effect as much as possible, we propose a novel and effective representation learning method for KGs named TransRHS. Our TranRHS model is elaborately designed to encode the RHS information so that this information can be seamlessly integrated into the embedding without affecting the original model. More specifically, for each triple (h, r, t) , $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ are learned for entities, and $\mathbf{r} \in \mathbb{R}^d$ together with a multi-dimensional relation-specific sphere are also learned. The information in RHS is encoded into knowledge graph embeddings by using relative positions among the vectors and the spheres. Then TransRHS is optimized by minimizing the corresponding loss functions.

In summary, we highlight our key contributions as follows,

1. To the best of our knowledge, we are the first to investigate the problem of representation learning of KGs with relation hierarchical structure, i.e., the *subRelationOf* generalization relationships among relations.
2. We propose a novel representation learning method named TransRHS, which merely adds a *relation-specific sphere* to the representation of each relation in vector space, but is capable of fusing the information of RHS into the KG embeddings.
3. The experimental results on link prediction and triple classification show that TransRHS outperforms all baselines by more than 20% and 5%, respectively, which verifies that RHS can significantly enhance the quality of knowledge graph embeddings.

2 Related Work

2.1 KG Embedding with Facts

As a pioneering work of knowledge graph embedding, TransE [Bordes *et al.*, 2013] opens a line of translation-based methods. TransE encodes both entities and relations as vectors in the same space, say \mathbb{R}^d . Given a fact (h, r, t) , the relation is interpreted as a translation vector \mathbf{r} so that the entities \mathbf{h} and \mathbf{t} can be linked by \mathbf{r} with low error, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds.

Despite its simplicity and efficiency, TransE still has drawbacks in dealing with 1-to-N, N-to-1, and N-to-N relations. For a 1-to-N relation, e.g., *directorOf*, TransE might learn very similar vector representations for *Titanic* and *Avatar* which are all films directed by *JamesCameron*, even though they are different entities.

To fix this issue of TransE, an effective strategy is to allow an entity to have distinct representations when involved in different relations. Following this general idea, TransH

[Wang *et al.*, 2014b] models entities again as vectors, but each relation r as a hyperplane together with a translation operation on that hyperplane, enabling an entity to play distinct roles in different scenarios. Furthermore, TransR [Lin *et al.*, 2015b] directly models entities and relations in separate entity and relation spaces, projecting entities from entity space to relation-specific space when judging the distance between entities. TransD [Ji *et al.*, 2015] simplifies TransR by further decomposing the projection matrix into a product of vectors. RotatE [Sun *et al.*, 2019] proposes a rotation-based translational method, which represents entities as complex vectors and relations as rotations in a complex vector space.

Another category of knowledge representation learning models is the semantic matching models which have also turned out to be effective. RESCAL [Nickel *et al.*, 2011] associates each entity with a vector to capture its latent semantics, and each relation is represented as a matrix which models pairwise interactions between latent factors. DistMult [Yang *et al.*, 2015] simplifies RESCAL by restricting relation matrices to diagonal matrices. However, this oversimplified model can only deal with symmetric relations that is clearly not powerful enough for general knowledge graphs. HolE [Nickel *et al.*, 2016] combines the expressive power of RESCAL with the efficiency and simplicity of DistMult. ComplEx [Trouillon *et al.*, 2016] extends DistMult by introducing complex-valued embeddings so as to better model asymmetric relations.

2.2 KG Embedding with Multi-Source Information

It has been widely recognized that multi-source information including entity types, relation paths, and textual descriptions, considered as supplements for the structured information implied in triples, is significant for KG embeddings.

The type information is available in most KGs, usually represented in a type of “is-a” relations. A straightforward method [Nickel *et al.*, 2012] to model type information is to regard the triples describing type information as ordinary triples. [Guo *et al.*, 2015] proposed semantically smooth embedding (SSE), which requires entities of the same type to stay close to each other in the embedding space. TKRL [Xie *et al.*, 2016b] is a translational distance model with type-specific entity projections, which can handle hierarchical entity categories and multiple category labels.

In most KGs, the textual information is another kind of common source that can be used for representation learning of KGs. KG embedding with textual information dates back to the NTN model [Socher *et al.*, 2013], where textual information is simply used to initialize entity representations. However, this method separates textual information separately from KG facts, and hence fails to leverage interactions between them. The first joint model [Wang *et al.*, 2014a], which aligns the given KG with an auxiliary text corpus, and then jointly conducts KG embedding and word embedding, makes better use of textual information during embedding. DKRL [Xie *et al.*, 2016a] proposes description-based representations for entities with CBOW or CNN, which is capable of modeling entities in the zero-shot scenario.

Rich information implied in relation hierarchical structure

is also significant for KGs, which has been surprisingly ignored. [Lin *et al.*, 2015a] presents PTransE, which encodes relation paths to embed both entities and relations in a low-dimensional space. [Zhang *et al.*, 2018] proposes a three-layer structure of relations, which is merely an abstract structure obtained by gathering similar relations into clusters of relations and splitting coarse-grained relations into several fine-grained sub-relations. Our RHS is a structure constructed by the generalization relationship named **subRelationOf**, which contains abundant semantic information. To take advantage of RHS, we propose a KG embedding model called TransRHS.

3 Our Approach

In this section, the technical details of how to take advantage of the information implied in the relation hierarchical structure are specified. First, the formal definitions of RHS and the problem formulation for representation learning of KGs with RHS are provided. Then, our method TransRHS is proposed and explained in detail. Finally, the new loss functions and the optimization method are introduced.

3.1 Relation Hierarchical Structure

Relation hierarchical structure, which implies inherent relevance among relations, is of significance for representation learning of KGs. This kind of information is usually encoded by a generalization relationship named **subRelationOf** between relations and stored in the form of triples. Taking the relations in Figure 1 for instance, a **subRelationOf** triple (athLedTeam, **subRelationOf**, athPlaysForTeam) indicates the semantic relevance between the relation athLedTeam and the relation athPlaysforTeam, that is, the former is a sub-relation of the latter.

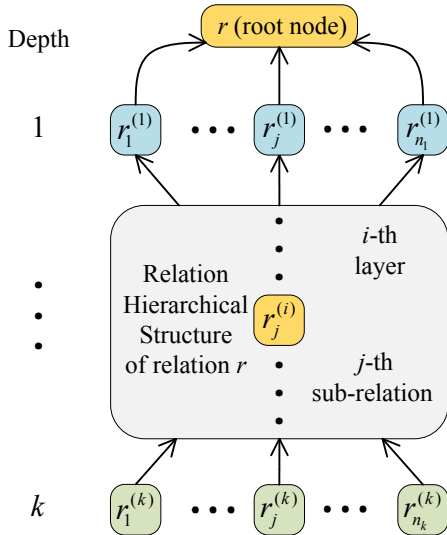


Figure 2: Formalization of relation hierarchical structure

As shown in Figure 2, a typical relation hierarchical structure is tree-like. We usually use the most general relation r as the root node of the corresponding tree structure, and $r_j^{(i)}$ represents the j -th sub-relation at the i -th layer in the relation hierarchical structure. n_i denotes the total number of sub-relations in the i -th layer. For simplicity, we use $r^{(k)}$ to represent a sub-relation at the k -th layer. A path from the root node r to any leaf node $r^{(k)}$ can be formalized as $P_r^k = \{r, r^{(1)}, r^{(2)}, \dots, r^{(k)}\}$, where the subscript r of P_r^k denotes the root node of this tree structure and the superscript k of that denotes the total number of sub-relations in this path.

Given a knowledge graph, it can be formalized as $\mathcal{G} = (E, R, T)$, where E represents the entity set, R the set of relations, and T the set of triple facts. Specifically, the relation set of the KG with RHS can be further formalized as $R = \{sro\} \cup R_l$, where R_l represents the set of relations between entities, and sro (i.e., **subRelationOf**) represents the generalization relationship between relations.

Therefore, the triple set T in a KG can be divided into two corresponding disjoint subsets:

1. **subRelationOf** triple set $T_r = \{(r, sro, p) \mid r, p \in R_l\}$, where r, p are relations between entities and sro is the generalization relationship between relations, namely **subRelationOf**.
2. Relational triple set $T_e = \{(h, r, t) \mid h, t \in E \wedge r \in R\}$, where h, t represent head entity and tail entity, respectively, and r represents a relation between entities.

Representation learning of KGs with RHS aims to learn embeddings for entities and relations in the same space \mathbb{R}^d . For each entity $e \in E$, we learn a low-dimensional vector $e \in \mathbb{R}^d$. For each relation $r \in R_l$ and its RHS information, we learn a low-dimensional vector $r \in \mathbb{R}^d$ together with a relation-specific sphere s . We typically use Δ to represent the set of positive triples. A KG embedding model should satisfy the following two formulas to incorporate the RHS information, namely

$$(h, r, t) \in T_e \wedge (r, sro, p) \in T_r \rightarrow (h, p, t) \in \Delta \quad (1)$$

and

$$(r, sro, p) \in T_r \wedge (p, sro, pp) \in T_r \rightarrow (r, sro, pp) \in \Delta \quad (2)$$

, which are essentially a realization of the *rdfs5* and *rdfs7* rules in the RDFS semantics, respectively [Hayes and Patel-Schneider, 2014].

3.2 TransRHS

Notations are described first. Given a positive fact (h, r, t) , h denotes a head entity, r represents a relation, and t denotes a tail entity. The bold letters \mathbf{h} , \mathbf{r} , and \mathbf{t} denote the corresponding embeddings. Δ denotes the set of positive triples, and Δ' represents the set of negative triples. Therefore, we use $(h, r, t) \in \Delta$ to indicate that (h, r, t) is a positive triple. TransE assumes that the score function

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} \quad (3)$$

is low if $(h, r, t) \in \Delta$, which indicates that the lower $f_r(h, t)$ is, the more probable (h, r, t) is to be a positive triple.

To take advantage of the implicit information implied in RHS, a novel method named TransRHS is proposed, where Formula (1) and Formula (2) can be explicitly embodied in embeddings. Taking Formula (1) for instance, when $(r, sro, p) \in T_r$ holds, if $(h, r, t) \in T_e$ is a positive triple, then (h, p, t) must be a positive triple. So the embeddings should satisfy that if $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, then $\mathbf{h} + \mathbf{p} \approx \mathbf{t}$. Furthermore, $f_p(h, t)$ should be less than $f_r(h, t)$ according to the Formula (3).

In order to encode the RHS information without increasing the complexity in both space and time, TranRHS merely introduces one more sphere for each relation compared with the TransE. For purpose of reducing the impact on the translation-based model to the minimum, TransRHS will follow the translational requirement, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) holds. In TransRHS, each entity $e \in E$ is learned as a low-dimensional vector $\mathbf{e} \in \mathbb{R}^d$ and each relation $r \in R_l$ is encoded as a low-dimensional vector $\mathbf{r} \in \mathbb{R}^d$ together with a *relation-specific sphere*. More specifically, given a pair of triples $(h, r, t) \in T_e$ and $(r, sro, p) \in T_r$, TransRHS learns relation p as a low-dimensional vector $\mathbf{p} \in \mathbb{R}^d$ and its corresponding spheres s_1 with radius m_1 . Analogously, $\mathbf{r} \in \mathbb{R}^d$ and the sphere s_2 with radius m_2 are obtained for relation r . And head entity h and tail entity t are represented as low-dimensional vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$. The distance between the $\mathbf{h} + \mathbf{p}$ and \mathbf{t} can be defined as:

$$d_1 = \|\mathbf{h} + \mathbf{p} - \mathbf{t}\|_2 \quad (4)$$

Similarly, we define the distance between $\mathbf{h} + \mathbf{r}$ and \mathbf{t} as:

$$d_2 = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2 \quad (5)$$

The ideal embeddings in TransRHS are shown in Figure 3 and meets the following conditions:

1. $\mathbf{h} + \mathbf{p}$ is inside the sphere s_1 ($d_1 < m_1$).
2. $\mathbf{h} + \mathbf{r}$ is between the sphere s_1 and s_2 ($m_1 < d_2 < m_2$).
3. The sphere s_1 is inside the sphere s_2 ($m_1 < m_2$).

The following situations still need to be optimized:

1. $\mathbf{h} + \mathbf{p}$ is outside the sphere s_1 ($d_1 > m_1$).
2. $\mathbf{h} + \mathbf{r}$ is outside the sphere s_2 ($d_2 > m_2$).
3. $\mathbf{h} + \mathbf{r}$ is inside the sphere s_1 ($d_2 < m_1$).
4. The sphere s_2 is inside the sphere s_1 ($m_1 > m_2$).

For the above conditions, we define the loss functions as:

$$f_{RHS} = \begin{cases} \alpha_1 [\|\mathbf{h} + \mathbf{p} - \mathbf{t}\|_2 - m_1]_+ & \text{if } d_1 > m_1 \\ \alpha_2 [\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2 - m_2]_+ & \text{if } d_2 > m_2 \\ \alpha_3 [m_1 - \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2]_+ & \text{if } d_2 < m_1 \\ \alpha_4 [m_1 - m_2]_+ & \text{if } m_1 > m_2 \end{cases} \quad (6)$$

The condition discussed above involves two layers of RHS. In fact, our model could be applied to encode the information

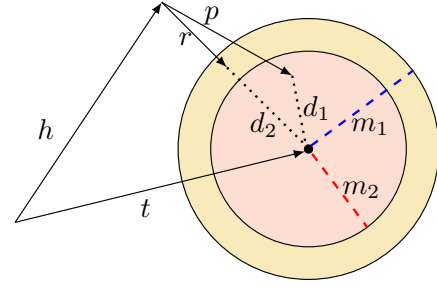


Figure 3: Illustration of TransRHS

of n -layer RHS into embeddings naturally. For the sake of illustration, we take a three-layer RHS as an example. Given a relational triple (h, r, t) and two subRelationOf triples (r, sro, p) and (p, sro, pp) , the spheres s_1, s_2 , and s_3 corresponding to the relations pp, p , and r are learned respectively. Similarly, the ideal embeddings should satisfy the following conditions:

1. $\mathbf{h} + \mathbf{pp}$ is inside the sphere s_1 .
2. $\mathbf{h} + \mathbf{p}$ is between the sphere s_1 and s_2 .
3. $\mathbf{h} + \mathbf{r}$ is between the sphere s_2 and s_3 .
4. The sphere s_1 is inside the sphere s_2 which is inside the sphere s_3 .

Based on the above analysis, our method is easily extended to be appropriate for encoding an n -layer RHS.

The above assumption indicates that the deeper the relation r located in the RHS, the larger the radius m of the sphere corresponding to the relation r . However, the increasing radius of the sphere will make $\mathbf{h} + \mathbf{r}$ to lie farther away from \mathbf{t} , which will conflict the basic principles of the translation-based models, i.e. $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. To address the above issue, we set the difference between the radii of two adjacent spheres should decrease as the depth of relations increases in RHS. We use $m_{(i)}$ to represent the radius of the relation at the i -th layer in RHS and design a proportional-declined weighting strategy among $m_{(i-1)}, m_{(i)}$, and $m_{(i+1)}$ as follows:

$$m_{(i)} - m_{(i-1)} : m_{(i+1)} - m_{(i)} = \eta : (1 - \eta) \quad (7)$$

in which we set $\eta \in (0.5, 1)$. This strategy indicates that the difference between $m_{(i)}$ and $m_{(i-1)}$ is larger than the difference between $m_{(i)}$ and $m_{(i+1)}$, which fits in our objective.

3.3 Training Method

For relational triples, we use Δ and Δ' to denote the positive triple set and the negative triple set, respectively. Then we can define a margin-based loss function for relational triples:

$$L_{ORI} = \sum_{(h,r,t) \in \Delta} \sum_{(h',r',t') \in \Delta'} [\gamma + f_r(h, t) - f_r(h', t')]_+ \quad (8)$$

where $[x]_+ \triangleq \max(0, x)$ and γ is the margin separating positive triples and negative triples.

	Relations	Inferences
Sport	athleteLedSportsTeam	$(x, \text{athleteLedSportsTeam}, y) \mapsto (x, \text{athletePlaysForTeam}, y)$
	athletePlaysForTeam	$(x, \text{athletePlaysForTeam}, y) \mapsto (x, \text{personBelongsToOrganization}, y)$
	coachesTeam	$(x, \text{coachesTeam}, y) \mapsto (x, \text{personBelongsToOrganization}, y)$
	personBelongsToOrganization	
Location	capitalCityOfCountry	$(x, \text{capitalCityOfCountry}, y) \mapsto (x, \text{cityLocatedInCountry}, y)$
	cityLocatedInCountry	$(x, \text{cityLocatedInCountry}, y) \mapsto (y, \text{cityLocatedInState}, x)$
	cityLocatedInState	
	stateHasCapital	
	stateLocatedInCountry	

Table 1: Relations and inferences in Sport and Location datasets.

The loss function for `subRelationOf` triples is defined as:

$$L_{RHS} = \sum_{(r, s, o, p) \in T_r} \sum_{(h, r, t) \in T_e} f_{RHS}(h, r, t, p) \quad (9)$$

The total loss of the TransRHS model is comprised of two parts, as is shown in Equation 10,

$$L = L_{ORI} + L_{RHS} \quad (10)$$

The goal of training TransRHS is to minimize the above loss functions, and iteratively update embeddings of entities, relations, and spheres.

4 Experiments

Our method is evaluated on two typical tasks commonly used in knowledge graph embedding: link prediction [Bordes *et al.*, 2013] and triple classification [Socher *et al.*, 2013].

4.1 Datasets and Experiment Settings

In this research, we evaluate our models on benchmarks Sport and Location, which are introduced by [Wang *et al.*, 2015] and extracted from NELL [Carlson *et al.*, 2010]. The relations in Sport and Location, along with their inferences, are listed in Table 1. Table 2 gives a summary of these datasets.

Considering that in most KGs there are no negative triples which are indispensable for the experiments, the process of model training involves generating negative triples from positive triples. There are two main strategies for generating negative triples: “bern” and “unif”. The former method simply obtains negative triples by randomly corrupting the positive triples. Taking a positive fact (h, r, t) for instance, a negative triple (h', r, t') or (h, r, t') is obtained by randomly sampling h' or t' from entity set. However, as a real KG is often far from complete, this way of randomly sampling may introduce many false negative triples into training. The latter strategy considers the mapping property of the relation, i.e., 1-to-N, N-to-1, and N-to-N, and sets different probabilities for replacing head or tail entity during corrupting triples.

TransRHS model is trained with mini-batch SGD. As for parameters, we select the batch size B among $\{20, 40, 80, 160\}$ and margin γ among $\{0.5, 1.0, 1.5, 2.0\}$. We also set the dimensions of entity and relation to be the same d . For the learning rate λ , we select a fixed-rate following [Bordes *et al.*, 2013]. The optimal configurations of our models are: $B = 40$, $\gamma = 1.0$, $\lambda = 0.001$.

Dataset	$ E $	$ R $	#Train	#Valid	#Test
Sport	1039	4	1349	358	358
Location	445	5	384	65	65

Table 2: Statistics on the datasets

For a fair comparison, all models are trained under the same dimension $d = 100$. For the comparison methods, we use the open source implementations in OpenKE [Han *et al.*, 2018] with default hyperparameters.

Compared with TransE, we merely introduce a *relation-specific sphere* for each relation. Therefore, the space and time complexity of TransRHS is the same as TransE, which scales linearly with the dimension d . Given a KG with n entities and m relations, the space complexity of TransRHS is $O(nd + md)$ and the time complexity is $O(d)$.

4.2 Link Prediction

Evaluation

Link prediction is to complete a triple (h, r, t) with h or t missing, i.e., predict t given (h, r) or predict h given (r, t) . Rather than requiring one best answer, this task emphasizes more on ranking a set of candidate entities from the knowledge graph. We follow the same protocol in TransE: for each testing triple (h, r, t) , we replace the tail t by every entity $e \in E$ in the knowledge graph and calculate a dissimilarity score (according to the scoring function f_r) on the corrupted triple (h, r, e) . Ranking the scores in ascending order, we then get the rank of the original correct triple. Similarly, we can get another rank for (h, r, t) by corrupting the head h . Let $\text{rank}_h(h, r, t)$ be the ranking of (h, r, t) among all head-corrupted relations and $\text{rank}_t(h, r, t)$ denotes a similar ranking with tail corruptions.

We use two evaluation metrics: Hits@N and Mean Reciprocal Rank (MRR). MRR is the mean of the reciprocal rank:

$$\text{MRR} = \frac{1}{2 * |T|} \sum_{(h, r, t) \in T} \frac{1}{\text{rank}_h(h, r, t)} + \frac{1}{\text{rank}_t(h, r, t)} \quad (11)$$

Hits@N measures the proportion of triples in T that rank among top t after corrupting both heads and tails. The above is called the “raw” setting. Notice that if a corrupted triple exists in the knowledge graph, ranking it before the test triple will have a bad effect on the experimental results. To eliminate this factor, we remove those corrupted triples which exist

Model	Sport					Location				
	MRR		Hits@N(%)			MRR		Hits@N(%)		
	Filter	Raw	1	3	10	Filter	Raw	1	3	10
TransE [Bordes <i>et al.</i> , 2013]	0.644	0.417	53.3	73.8	80.0	0.548	0.492	44.6	63.8	69.2
TransH [Wang <i>et al.</i> , 2014b]	0.540	0.368	40.2	65.5	73.3	0.421	0.375	26.1	53.8	63.8
TransR [Lin <i>et al.</i> , 2015b]	0.423	0.304	28.0	52.6	65.6	0.553	0.489	49.2	59.2	65.3
TransD [Ji <i>et al.</i> , 2015]	0.443	0.319	27.2	57.6	69.5	0.162	0.144	10.0	15.3	24.6
RotatE [Sun <i>et al.</i> , 2019]	0.685	0.441	61.3	73.8	80.4	0.402	0.363	36.1	40.7	47.6
TransE-HRS [Zhang <i>et al.</i> , 2018]	0.661	0.429	58.5	70.9	79.4	0.414	0.380	36.9	42.3	49.2
RESCAL [Nickel <i>et al.</i> , 2011]	0.071	0.058	4.0	7.9	12.2	0.183	0.166	7.6	20.0	40.7
DistMult [Yang <i>et al.</i> , 2015]	0.463	0.341	39.8	49.1	58.3	0.346	0.328	30.7	35.3	41.3
HolE [Nickel <i>et al.</i> , 2016]	0.269	0.184	22.4	26.8	35.4	0.356	0.330	33.0	36.1	39.2
ComplEx [Trouillon <i>et al.</i> , 2016]	0.511	0.365	43.0	55.1	66.2	0.365	0.342	33.0	36.9	43.0
Simple [Kazemi and Poole, 2018]	0.495	0.380	42.7	52.0	62.8	0.357	0.331	31.5	37.6	40.7
TransRHS (unif)	0.820	0.515	80.5	82.8	84.2	0.652	0.572	63.1	66.2	69.2
TransRHS (bern)	0.809	0.514	78.7	82.1	83.7	0.660	0.586	64.6	66.2	69.8

Table 3: Evaluation results on link prediction. Best results are in bold.

in either train, valid, or test set before getting the rank of each test triple. This setting is called “filter”.

Results

The results of the link prediction are shown in Table 3. From the results, we observe that: (1) TransRHS significantly outperforms all baselines in terms of both MRR and Hits@N, which indicates that the RHS information is successfully encoded into entity and relation embeddings and could improve the representation learning of knowledge graphs. (2) The “unif” sampling trick works well for Sport and the “bern” sampling trick behaves well for Location.

4.3 Triple Classification

Evaluation

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

Negative triples are required for the evaluation of triple classification. Hence, we construct some negative triples fol-

lowing the same setting in [Socher *et al.*, 2013]. There are as many true triples as negative triples in both valid and test set.

The classification strategy is conducted as follows: We set different relation-specific thresholds δ_r for each relation. For a triple (h, r, t) , if the dissimilarity score of $f_r(h, t)$ is below δ_r , the triple is then predicted to be positive and otherwise negative. The relation-specific thresholds δ_r are optimized by maximizing the classification accuracies in all triples with the corresponding r on the valid sets.

Results

We evaluate this task on both Sport and Location. Evaluation results on triple classification are shown in Table 4. From Table 4 we observe that: (1) The empirical results in triple classification demonstrate that TransRHS outperforms all baselines, which once again confirms the conclusions we obtained from link prediction and indicates that TransRHS is capable of encoding RHS information into knowledge graph embeddings. (2) “bern” sampling trick works well for both datasets.

5 Conclusion

In this paper, we propose a novel model named TransRHS for representation learning of knowledge graphs with relation hierarchical structure. More specifically, each relation in knowledge graph is encoded into a low-dimensional vector together with a relation-specific sphere in the same space. In experiments, we evaluate our model on two typical tasks including link prediction and triple classification. Experimental results show that RHS information is significant for representation learning of knowledge graphs, and TransRHS model is capable of encoding RHS information into KG embeddings.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61972275) and the Australian Research Council Linkage Project (LP180100750).

Model	Accuracy(%)	
	Sport	Location
TransE [Bordes <i>et al.</i> , 2013]	89.2	68.4
TransH [Wang <i>et al.</i> , 2014b]	86.8	73.8
TransR [Lin <i>et al.</i> , 2015b]	85.6	66.9
TransD [Ji <i>et al.</i> , 2015]	86.1	60.7
RotatE [Sun <i>et al.</i> , 2019]	89.2	72.3
TransE-HRS [Zhang <i>et al.</i> , 2018]	86.9	69.7
RESCAL [Nickel <i>et al.</i> , 2011]	57.8	57.6
DistMult [Yang <i>et al.</i> , 2015]	85.7	58.4
HolE [Nickel <i>et al.</i> , 2016]	77.3	69.2
ComplEx [Trouillon <i>et al.</i> , 2016]	86.3	74.6
Simple [Kazemi and Poole, 2018]	86.5	67.6
TransRHS (unif)	89.8	74.8
TransRHS (bern)	90.9	78.4

Table 4: Evaluation results on triple classification

References

- [Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
- [Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
- [Carlson *et al.*, 2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*, 2010.
- [Guo *et al.*, 2015] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. Semantically smooth knowledge graph embedding. 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)*, pages 84–94, 2015.
- [Han *et al.*, 2018] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, 2018.
- [Hayes and Patel-Schneider, 2014] Patrick J Hayes and Peter F Patel-Schneider. Rdf 1.1 semantics. *W3C recommendation*, 25:7–13, 2014.
- [Ji *et al.*, 2015] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, 2015.
- [Kazemi and Poole, 2018] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, pages 4284–4295, 2018.
- [Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of EMNLP*, pages 705–714, 2015.
- [Lin *et al.*, 2015b] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816, 2011.
- [Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280, 2012.
- [Nickel *et al.*, 2016] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Thirtieth Aaai conference on artificial intelligence*, 2016.
- [Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.
- [Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations*, 2019.
- [Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML), 2016.
- [Wang *et al.*, 2014a] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1591–1601, 2014.
- [Wang *et al.*, 2014b] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.
- [Wang *et al.*, 2015] Quan Wang, Bin Wang, and Li Guo. Knowledge base completion using embeddings and rules. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [Xie *et al.*, 2016a] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [Xie *et al.*, 2016b] Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, pages 2965–2971, 2016.
- [Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations*, 2015.
- [Zhang *et al.*, 2018] Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3198–3207, 2018.