

Exploiting Global Semantic Similarities in Knowledge Graphs by Relational Prototype Entities

Xueliang Wang, Jiajun Chen, Feng Wu, Jie Wang

University of Science and Technology of China

Hefei, Anhui, China

{xlwang95,jiajun98}@mail.ustc.edu.com,{fengwu,jiewangx}@ustc.edu.cn

ABSTRACT

Knowledge graph (KG) embedding aims at learning the latent representations for entities and relations of a KG in continuous vector spaces. An empirical observation is that the head (tail) entities connected by the same relation often share similar semantic attributes—specifically, they often belong to the same category—no matter how far away they are from each other in the KG; that is, they share global semantic similarities. However, many existing methods derive KG embeddings based on the local information, which fail to effectively capture such global semantic similarities among entities. To address this challenge, we propose a novel approach, which introduces a set of virtual nodes called **relational prototype entities** to represent the prototypes of the head and tail entities connected by the same relations. By enforcing the entities' embeddings close to their associated prototypes' embeddings, our approach can effectively encourage the global semantic similarities of entities—that can be far away in the KG—connected by the same relation. Experiments on the entity alignment and KG completion tasks demonstrate that our approach significantly outperforms recent state-of-the-arts.

CCS CONCEPTS

• Information systems → Data mining; • Computing methodologies → Natural language processing.

KEYWORDS

knowledge graph embedding, relational prototype entities, entity alignment, knowledge graph completion

ACM Reference Format:

Xueliang Wang, Jiajun Chen, Feng Wu, Jie Wang. 2020. Exploiting Global Semantic Similarities in Knowledge Graphs by Relational Prototype Entities. In *KDD 2020: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 22–27, 2020, San Diego, California USA*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Knowledge graphs (KGs) store a wealth of human knowledge in a structured way, which are usually collections of the factual triple:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD 2020, August 22–27, 2020, San Diego, California USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

(*head entity, relation, tail entity*). A large number of real-world KGs, such as Freebase [1], DBpedia [18], and YAGO [33], have been created and successfully applied to a wide range of applications, such as information retrieval [47], recommendation systems [40, 42], question answering [12], etc.

Although commonly used KGs contain billions of triples, a single KG is usually far from complete and may not support these applications with sufficient facts [10]. To tackle this issue, two KG tasks know as *entity alignment* and *KG completion* are proposed and have achieved massive attention recently [2, 8, 10, 13, 26, 32, 45, 48, 50]. Entity alignment, a.k.a., entity resolution or matching, aims to identify entities from different KGs that refer to the same real-world object. KG completion, a.k.a., link prediction, aims to automatically predict missing links between entities based on known links.

Inspired by word embedding [25] that can well capture semantic meaning of words, researchers have paid increasing attention to *knowledge graph embedding* [2, 4, 19, 21, 27, 28, 34, 43]. The key idea of KG embedding is to embed entities and relations of a KG into continuous vector spaces, so as to simplify the manipulation while preserving the inherent structures of the KG [41]. For example, the distance-based models, such as TransE [2], TransH [44], TransR [20], and RotatE [34], measure the plausibility of a fact as the distance between the two entities after an operation (e.g., translation or rotation) carried out by the relation. Moreover, the semantic matching models, such as RESCAL [28], DistMult [49], HolE [27], ComplEx [39] and ConvE [7], measure plausibility of facts by matching latent representations of entities and relations embodied in their vector spaces. More recently, the GCN-based models, such as R-GCN [31], GCN-Align [43], MuGNN [3], and AliNet [37], are proposed to make use of the structure information in the KG, where the embedding of an entity is learned by recursively aggregating the embeddings of its neighbor entities.

While promising, many existing KG embedding models derive embeddings by using the local information of entities in the KG. They learn the embedding of an entity by aggregating the embeddings of its one-hop neighbors (e.g., TransE and RESCAL) or multi-hop neighbors (e.g., multi-layer GCNs). An empirical observation is that the head (tail) entities connected by the same relation often share similar semantic attributes—specifically, they often belong to the same category—no matter how far away they are from each other in the KG; that is, they share global semantic similarities [11, 17, 22, 46]. For example, given three triples: (*Sydney Pollack, directed, ?*), (*Franco Zeffirelli, directed, ?*), and (*Yasuo Fukuda, isPoliticianOf, ?*), the former two predicted entities should belong to the category of “film”, thereby the embeddings of them should be more similar. However, many existing KG embedding methods do not

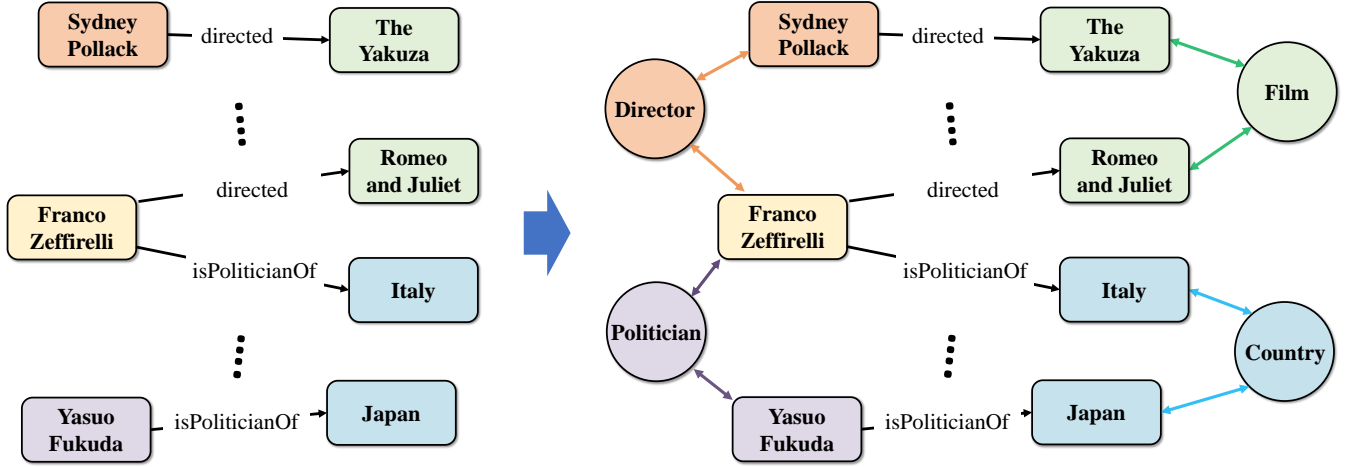


Figure 1: Illustration of our proposed method. The left is the original knowledge graph, and the right is the knowledge graph with relational prototype entities. In our method, we connect the head (tail) entities connected by the same relation to the associated head (tail) relational prototype entities. In the figure, the square blocks represent entities, and the round blocks represent relational prototype entities. Notice that, the entity *Franco Zeffirelli* is connected to two relational prototype entities at the same time.

explicitly model such global semantic similarities among entities, which results in that the embeddings are not expressive enough.

To tackle the aforementioned challenge, we innovatively introduce a set of virtual nodes called **relational prototype entities**. Specifically, for each relation, we define a head (tail) category for the head (tail) entities connected by this relation; that is, if we have N relations in the KG, we will define N head categories and N tail categories, respectively. The proposed relational prototype entities represent the prototypes for each category. We then connect the entities from the same category—no matter how far away they are from each other in the KG—to the associated relational prototype entities. In this way, the proposed relational prototype entities can serve as the junctions to enhance the interaction of entities’ embeddings from the same category. Notice that, an entity may be connected to more than one relational prototype entity. For example, given two triples: *(Franco Zeffirelli, directed, Romeo and Juliet)* and *(Franco Zeffirelli, isPoliticianOf, Italy)*, the head entity *Franco Zeffirelli*—which belongs to the categories of “director” and “politician”—will be connected to the head relational prototype entities of *directed* and *isPoliticianOf* at the same time. We show an illustration in Figure 1.

Our proposed method will enforce the entities’ embeddings close to their associated prototypes’ embeddings, so as to effectively encourage the global semantic similarities of entities—that can be far away in the KG—connected by the same relation. In this way, the embeddings in the KG can be more expressive, especially for the long-tail entities [10] whose local information is not rich. To show the effectiveness of our proposed method, we equip the popular GCN [16] and RotatE [34] models with our method and apply them to the entity alignment and KG completion tasks, respectively. Experiments show that our proposed method can not only effectively capture the global semantic similarities in the KG, but also

significantly outperform recent state-of-the-art methods on the benchmark datasets.

2 NOTATIONS AND BACKGROUND

We first introduce the notations used in the paper in Section 2.1. Then, we review the background of the entity alignment and KG completion tasks in Sections 2.2 and 2.3, respectively.

2.1 Notations

In this paper, we formally represent a KG as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where \mathcal{E} is the set of entities, \mathcal{R} is the set of relations, and $\mathcal{T} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of relational triples. We use the triple (h, r, t) to denote a fact in the KG, where the lower-case letters h , r , and t denote head entities, relations, and tail entities, respectively. The boldface lower-case letters \mathbf{h} , \mathbf{r} , and \mathbf{t} represent the corresponding vectors, and we use $[\mathbf{h}]_i$ to denote the i -th entry of the vector \mathbf{h} . Let k denote the embedding dimension. Given a set \mathcal{S} , we use $|\mathcal{S}|$ to denote the number of elements in \mathcal{S} . We use $\|\cdot\|$ to denote the Euclidean norm. Let $\circ : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ denote the Hadamard product between two vectors, i.e.,

$$[\mathbf{a} \circ \mathbf{b}]_i = [\mathbf{a}]_i \cdot [\mathbf{b}]_i \quad (1)$$

2.2 Entity Alignment

The entity alignment task aims to find entities from different KGs that refer to the same real-world object. Formally, given two KGs, i.e., $\mathcal{G}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1)$ and $\mathcal{G}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2)$, the goal of this task is to find alignment of entities between \mathcal{G}_1 and \mathcal{G}_2 based on the partial pre-aligned entity pairs $\mathcal{A}^+ = \{(i, j) \in \mathcal{E}_1 \times \mathcal{E}_2 | i \equiv j\}$, where \equiv represents the alignment relationship.

A critical challenge of the entity alignment task lies in the heterogeneity of structures because counterpart entities in different KGs usually have dissimilar neighborhood structures [30]. Recently,

the GCN-based embedding models have shown great potentials in dealing with the heterogeneity problem of the entity alignment task in both monolingual and cross-lingual scenarios [3, 37, 43]. The GCN-based models update the embedding of an entity by aggregating the embeddings of its neighbors, and the similarity of entities is measured by the distance of entity embeddings.

A popular variant of GCNs is the multi-layer vanilla GCN proposed by [16]. Every GCN encoder takes the hidden states of entity embeddings in the current layer as input, and outputs new entity embeddings with the following layer-wise propagation rule

$$\mathbf{e}_i^{(l)} = \rho \left(\frac{\sum_{j \in \mathcal{N}_1(i) \cup \{i\}} \mathbf{W}^{(l)} \mathbf{e}_j^{(l-1)}}{|\mathcal{N}_1(i) \cup \{i\}|} \right), \quad (2)$$

where $\mathbf{e}_i^{(l)}$ is the embedding of entity i at the l -th layer, $\mathcal{N}_1(i)$ is the set of one-hop entity neighbors of entity i , $\mathbf{W}^{(l)}$ is the trainable parameters at the l -th layer, and $|\mathcal{N}_1(i) \cup \{i\}|$ is the number of elements in the set $\mathcal{N}_1(i) \cup \{i\}$. ρ represents the activation function.

GCN-Align [43] uses the vanilla GCN [16] to embed entities of different KGs into a unified vector space, and entity alignments are discovered based on the distances between entities in the embedding space. Other extensions including GMNN [48], RDGCN [45] and AVR-GCN [50]. More recently, MuGNN [3] proposes a two-step method of rule-based KG completion and multi-channel GNNs for entity alignment. The learned rules rely on relation alignment to resolve schema heterogeneity. AliNet [37] proposes gated multi-hop neighborhood aggregation to mitigate the non-isomorphism of neighborhood structures in KGs. However, the most aforementioned models focus on the local structure information (e.g., one-hop or multi-hop neighbors), which do not explicitly consider the global semantic similarities among entities mentioned above.

2.3 KG Completion

The KG completion task aims to complete the missing facts in a single KG. Formally, given a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, it aims to predict the head entity h given $(?, r, t)$ or predict the tail entity t given $(h, r, ?)$. To tackle this problem, a general approach is to define a score function $f_r(\mathbf{h}, \mathbf{t})$ to measure the plausibility of each fact (h, r, t) . The goal of the optimization is to score true facts (h, r, t) observed in the KG higher than false facts (h', r, t) or (h, r, t') .

A straightforward method to tackle the KG completion task is to model $f_r(\mathbf{h}, \mathbf{t})$ as a distance-based function. For example, the popular TransE [2] and its extensions, such as TransH [44], TransR [20], TransD [13], and TransSparse [14], model relations as translations. More recently, the RotatE model [34] defines each relation as a rotation instead of a translation from the head entity to the tail entity in the complex vector space. The score function $f_r(\mathbf{h}, \mathbf{t})$ of RotatE is defined as

$$f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|, \quad (3)$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$, and the modulus of $[\mathbf{r}]_i$ is 1. Compared to the existing translational models, RotatE can better model symmetry relations [34]. Notice that, the aforementioned methods are triple-level learning methods [10], and most of them only consider the one-hop neighbors and do not explicitly model the global semantic

similarities that the head and tail entities connected by the same relation often belong to the same category, respectively.

3 METHODS

As mentioned above, many knowledge graph embedding methods mainly focus on using local information in the KG, which do not explicitly consider the global semantic similarities among entities. Inspired by the empirical observation that the head (tail) entities connected by the same relation often share similar semantic attributes, we innovatively introduce a set of relational prototype entities $P(r)$, and we use $\mathbf{P}(r)$ to represent the corresponding vector. Specifically, for each relation r , we define a head relational prototype entity $P_H(r)$ and a tail relational prototype entity $P_T(r)$ to serve as the prototypes for the head and tail categories of each relation. $\mathbf{P}_H(r)$ and $\mathbf{P}_T(r)$ are the corresponding vectors of $P_H(r)$ and $P_T(r)$. Then, we add the head and tail relational prototype entities to the KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ in a simple yet effective way. We connect the entities from the same category to the associated relational prototype entities. Thereby, the proposed relational prototype entities can serve as the junctions to enhance the interaction of entity embeddings from the same category. We show a simple illustration of our method in Figure 2.

The entity alignment and KG completion are two fundamental tasks in KG embedding [10]. To show the effectiveness and universality of our proposed approach, we equip the popular GCN [16] and RotatE [34] models with relational prototype entities and apply them to the entity alignment and KG completion tasks, respectively. In Sections 3.1 and 3.2, we detail the proposed methods for the two tasks. In Section 3.3, we analyze the proposed methods.

3.1 Entity Alignment

As introduced in Section 2.2, the GCN-based embedding models [3, 37, 43, 45, 50] have achieved great success on the entity alignment task recently. Therefore, we use the vanilla GCN [16] introduced in Section 2.2 as our baseline model. The baseline GCN gets the embedding of entity i by aggregating the embeddings of its one-hop entity neighbors by (2). In our proposed method, we add relational prototype entities to the KG. Thus, we redefine the propagation rule in (2) as

$$\mathbf{e}_i^{(l)} = \rho \left(\frac{\lambda \sum_{j \in \mathcal{N}_1(i) \cup \{i\}} \mathbf{W}^{(l)} \mathbf{e}_j^{(l-1)} + (1-\lambda) \sum_{P(r) \in \mathcal{N}_1^R(i)} \mathbf{W}^{(l)} \mathbf{P}^{(l-1)}(r)}{\lambda |\mathcal{N}_1(i) \cup \{i\}| + (1-\lambda) |\mathcal{N}_1^R(i)|} \right), \quad (4)$$

$$\mathbf{P}^{(l)}(r) = \rho \left(\frac{\lambda \sum_{j \in \mathcal{N}_1(P(r))} \mathbf{W}^{(l)} \mathbf{e}_j^{(l-1)} + (1-\lambda) \mathbf{W}^{(l)} \mathbf{P}^{(l-1)}(r)}{\lambda |\mathcal{N}_1(P(r))| + 1 - \lambda} \right), \quad (5)$$

where $\mathcal{N}_1(i)$ is the set of one-hop entity neighbors of entity i , $\mathcal{N}_1^R(i)$ is set of relational prototype entities directly connected to the entity i , and $\mathcal{N}_1(P(r))$ is the set of one-hop entity neighbors of the relational prototype entity $P(r)$. $0 < \lambda \leq 1$ is the hyperparameter that balances the weight. Notice that, when $\lambda = 1$, (4) is the same as (2); that is, the relational prototype entities will not affect the embeddings of entities in the original KG. In our method, the entities connected by the same relation can share global semantic

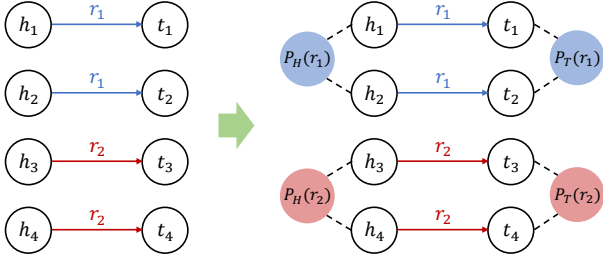


Figure 2: A simple illustration of our proposed method. $P_H(r_1)$ and $P_T(r_1)$ represent the head and tail relational prototype entities of relation r_1 , respectively. $P_H(r_2)$ and $P_T(r_2)$ represent the head and tail relational prototype entities of relation r_2 , respectively.

similarities through relational prototype entities, and we call this modified model **RPE-GCN**.

Many GCN-based methods use the hidden outputs at the last layer as the final embedding of each entity, i.e., $\mathbf{e}_i = \mathbf{e}_i^{(L)}$, where L is the number of GCN layers. However, the hidden representations of all layers could contribute to propagating alignment information [37]. Therefore, we get the embedding of each entity by aggregating the hidden states of all the layers following [37], i.e.,

$$\mathbf{e}_i = \frac{1}{L} \sum_{l=1}^L \mathbf{e}_i^{(l)}. \quad (6)$$

Loss Function. On the entity alignment task, we usually judge whether two entities are equivalent by the distance between them. We want the embeddings of aligned entities to have a small distance while those of unaligned entities have a larger distance. Following [3], the loss function is given as below

$$\mathcal{L} = \sum_{(i,j) \in \mathcal{A}^+} \sum_{(i',j') \in \mathcal{A}^-} [\|\mathbf{e}_i - \mathbf{e}_j\| + \gamma - \|\mathbf{e}_{i'} - \mathbf{e}_{j'}\|]_+, \quad (7)$$

where \mathcal{A}^- is the set of negative entity alignment pairs of \mathcal{A}^+ , $[\cdot]_+ = \max\{0, \cdot\}$ is the maximum between 0 and the input, and $\gamma > 0$ is the margin hyperparameter separating positive and negative entity alignments. Following [3, 36], we select 25 entities closest to the corresponding entity as negative samples by calculating cosine similarity in the same KG, and negative samples will be recalculated every 5 epochs.

3.2 KG Completion

As discussed in Section 2.3, many KG completion methods are triple-level learning methods [10], which fail to explicitly model the global semantic similarities among entities. To tackle the limitations, we redefine the entity embeddings by incorporating the embeddings of the associated relational prototype entities. We reformulate the score function used in the KG completion task as $f_r(g(\mathbf{h}, \mathbf{P}_H(r)), g(\mathbf{t}, \mathbf{P}_T(r)))$, where $g(\cdot)$ is the aggregation function.

In this paper, we choose the popular RotatE [34] as the baseline model in the experiments. We then redefine the score function in

(3) as:

$$f_r(g(\mathbf{h}, \mathbf{P}_H(r)), g(\mathbf{t}, \mathbf{P}_T(r))) = -\|(\lambda \mathbf{h} + (1 - \lambda) \mathbf{P}_H(r)) \circ \mathbf{r} - (\lambda \mathbf{t} + (1 - \lambda) \mathbf{P}_T(r))\|,$$

where $0 < \lambda \leq 1$ is the hyperparameter to balance the weight, and we choose the sum operation as the aggregation function. We call the modified model **RPE-RotatE**.

Loss Function. To train the model, we use the negative sampling loss functions with self-adversarial training [34]:

$$L = -\log \sigma(\gamma - f_r(g(\mathbf{h}, \mathbf{P}_H(r)), g(\mathbf{t}, \mathbf{P}_T(r)))) - \sum_{i=1}^n p(h'_i, r, t'_i) \log \sigma(f_r(g(\mathbf{h}'_i, \mathbf{P}_H(r)), g(\mathbf{t}'_i, \mathbf{P}_T(r))) - \gamma),$$

where $\gamma > 0$ is a fixed margin, σ is the sigmoid function, and (h'_i, r, t'_i) is the i -th negative triple. Moreover,

$$p(h'_j, r, t'_j | \{(h_i, r_i, t_i)\}) = \frac{\exp \alpha f_r(g(\mathbf{h}'_j, \mathbf{P}_H(r)), g(\mathbf{t}'_j, \mathbf{P}_T(r)))}{\sum_i \exp \alpha f_r(g(\mathbf{h}'_i, \mathbf{P}_H(r)), g(\mathbf{t}'_i, \mathbf{P}_T(r)))}$$

is the probability distribution of sampling negative triples, where α is the temperature of sampling.

3.3 Analysis

This section provides the theoretical analysis of our proposed method. We take the KG completion task as an example. We first prove that our method can enforce the entities' embeddings close to their associated prototypes' embeddings. Then, we prove that such property will benefit the performance. For the analysis of the entity alignment task, please refer to the Appendix A.

We begin with the assumption that the embeddings of different relational prototype entities are different. We then define the head and tail relational prototype areas $C_H(r)$ and $C_T(r)$ as

$$C_H(r) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{P}_H(r)\| \leq R_H(r)\}, \\ C_T(r) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{P}_T(r)\| \leq R_T(r)\},$$

where $R_H(r)$ and $R_T(r)$ are given by

$$R_H(r) = \max\{\|\mathbf{h} - \mathbf{P}_H(r)\| : (h, r, t) \in \mathcal{T}\}, \\ R_T(r) = \max\{\|\mathbf{t} - \mathbf{P}_T(r)\| : (h, r, t) \in \mathcal{T}\}.$$

The set of relational prototype areas is defined as

$$\mathcal{C} = \cup_{r \in \mathcal{R}} (\{C_H(r)\} \cup \{C_T(r)\}),$$

and the distance between $C_1, C_2 \in \mathcal{C}$ is

$$d(C_1, C_2) = \inf\{\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2\}.$$

In our method, we redefine the embedding of an entity by aggregating the embeddings of its associated relational prototype entity and itself. For the triple (h, r, t) , the aggregation operation $g(\cdot)$ is

$$\hat{\mathbf{h}} = g(\mathbf{h}, \mathbf{P}_H(r)) = \lambda \mathbf{h} + (1 - \lambda) \mathbf{P}_H(r),$$

where $\hat{\mathbf{h}}$ denotes the redefined embedding of \mathbf{h} , and $0 < \lambda \leq 1$ is the hyperparameter. It is clear that the distance between $\hat{\mathbf{h}}$ and $\mathbf{P}_H(r)$ becomes smaller than that between \mathbf{h} and $\mathbf{P}_H(r)$, i.e.,

$$\|\hat{\mathbf{h}} - \mathbf{P}_H(r)\| = \lambda \|\mathbf{h} - \mathbf{P}_H(r)\| \leq \|\mathbf{h} - \mathbf{P}_H(r)\|, \quad (8)$$

and the equal sign holds only when $\mathbf{h} = \mathbf{P}_H(r)$ or $\lambda = 1$.

The radii of the head and tail relational prototype areas after aggregation are

$$\begin{aligned}\widehat{R}_H(r) &= \max\{\lambda\|\mathbf{h} - \mathbf{P}_H(r)\| : (h, r, t) \in \mathcal{T}\} = \lambda R_H(r), \\ \widehat{R}_T(r) &= \max\{\lambda\|\mathbf{t} - \mathbf{P}_T(r)\| : (h, r, t) \in \mathcal{T}\} = \lambda R_T(r).\end{aligned}$$

Thus, we can conclude that every relational prototype area shrinks after the aggregation operation when $0 < \lambda < 1$. If the value of λ is small enough, there will be no overlap between any two different relational prototype areas in \mathcal{C} .

Next, we will prove that such property benefits the performance on the KG completion task. We begin with the assumption:

$$f_r(\mathbf{P}_H(r), \mathbf{P}_T(r)) = -\|\mathbf{P}_H(r) \circ \mathbf{r} - \mathbf{P}_T(r)\| = 0 \Rightarrow \mathbf{P}_H(r) \circ \mathbf{r} = \mathbf{P}_T(r).$$

LEMMA 3.1. *For a relation r , the following inequalities hold*

$$\begin{aligned}f_r(\mathbf{h}, \mathbf{t}) &\geq -\|\mathbf{h} - \mathbf{P}_H(r)\| - \|\mathbf{t} - \mathbf{P}_T(r)\|, \\ f_r(\mathbf{h}, \mathbf{t}) &\leq \|\mathbf{t} - \mathbf{P}_T(r)\| - \|\mathbf{h} - \mathbf{P}_H(r)\|, \\ f_r(\mathbf{h}, \mathbf{t}) &\leq \|\mathbf{h} - \mathbf{P}_H(r)\| - \|\mathbf{t} - \mathbf{P}_T(r)\|.\end{aligned}$$

PROOF. Based on the assumption $\mathbf{P}_H(r) \circ \mathbf{r} = \mathbf{P}_T(r)$, we can get

$$\begin{aligned}f_r(\mathbf{h}, \mathbf{t}) &= -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| = -\|(\mathbf{h} - \mathbf{P}_H(r)) \circ \mathbf{r} - (\mathbf{t} - \mathbf{P}_T(r))\| \\ &\geq -\|(\mathbf{h} - \mathbf{P}_H(r)) \circ \mathbf{r}\| - \|\mathbf{t} - \mathbf{P}_T(r)\|.\end{aligned}$$

As the modulus of $[\mathbf{r}]_i$ is 1, we have

$$\|(\mathbf{h} - \mathbf{P}_H(r)) \circ \mathbf{r}\| = \|\mathbf{h} - \mathbf{P}_H(r)\|.$$

Then, we can get

$$f_r(\mathbf{h}, \mathbf{t}) \geq -\|\mathbf{h} - \mathbf{P}_H(r)\| - \|\mathbf{t} - \mathbf{P}_T(r)\|.$$

We can prove the other two inequalities similarly. Thus, the lemma is proved. \square

Based on the conclusion that there will be no overlap between any two different relational prototype areas in \mathcal{C} if λ is small enough, we can get the following two theorems. Notice that, for convenience, \mathbf{h} and \mathbf{t} used in the following theorems represent the entity embeddings after the aggregation operation.

THEOREM 3.2. *For a relation r , if $d(C, C_H(r)) > 2R_T(r)$ for any $C \in \mathcal{C} \setminus \{C_H(r)\}$, we have $f_r(\mathbf{h}_1, \mathbf{t}) > f_r(\mathbf{h}_2, \mathbf{t})$ for any $\mathbf{h}_1 \in C_H(r)$, $\mathbf{h}_2 \notin C_H(r)$, $\mathbf{t} \in C_T(r)$.*

PROOF. As $\mathbf{h}_2 \notin C_H(r)$, there exists a $C_{h2} \in \mathcal{C} \setminus \{C_H(r)\}$ such that $\mathbf{h}_2 \in C_{h2}$. Then, we have

$$\|\mathbf{h}_2 - \mathbf{P}_H(r)\| \geq d(C_{h2}, C_H(r)) + R_H(r) > R_H(r) + 2R_T(r).$$

Based on Lemma 3.1, we can get

$$\begin{aligned}f_r(\mathbf{h}_1, \mathbf{t}) &\geq -\|\mathbf{h}_1 - \mathbf{P}_H(r)\| - \|\mathbf{t} - \mathbf{P}_T(r)\| \geq -R_H(r) - R_T(r), \\ f_r(\mathbf{h}_2, \mathbf{t}) &\leq \|\mathbf{t} - \mathbf{P}_T(r)\| - \|\mathbf{h}_2 - \mathbf{P}_H(r)\| < -R_H(r) - R_T(r).\end{aligned}$$

Therefore, we have $f_r(\mathbf{h}_1, \mathbf{t}) > f_r(\mathbf{h}_2, \mathbf{t})$, which completes the proof. \square

THEOREM 3.3. *For a relation r , if $d(C, C_T(r)) > 2R_H(r)$ for any $C \in \mathcal{C} \setminus \{C_T(r)\}$, we have $f_r(\mathbf{h}, \mathbf{t}_1) > f_r(\mathbf{h}, \mathbf{t}_2)$ for any $\mathbf{h} \in C_H(r)$, $\mathbf{t}_1 \in C_T(r)$, $\mathbf{t}_2 \notin C_T(r)$.*

PROOF. We omit the proof of Theorem 3.3 since it is similar to the proof of Theorem 3.2. \square

From Theorem 3.2 and Theorem 3.3, we can see that the score of entities belonging to the category of the correct answer will be higher than entities from other categories.

In conclusion, our method can enforce the entities' embeddings close to their associated prototypes' embeddings, so as to encourage the returned entities with high ranking to all belong to the category of the correct answer. For example, when predicting the director of a film, our method will encourage the score of entities belonging to the category of "director" to be higher than other entities, which is beneficial to improving the performance.

Table 1: Statistics of the number of relations, entities, and triples for DBP15k and DWY100k datasets.

Datases	#Relation	#Entity	#Triple
DBP _{ZH}	2,830	66,469	153,929
DBP _{EN}	2,317	98,125	237,674
DBP _{JA}	2,043	65,744	164,373
DBP _{EN}	2,096	95,680	233,319
DBP _{FR}	1,379	66,858	192,191
DBP _{EN}	2,209	105,889	278,590
DWY _{DB}	330	100,000	463,294
DWY _{WD}	220	100,000	448,774
DWY _{DB}	302	100,000	428,952
DWY _{YG}	31	100,000	502,563

4 EXPERIMENTS AND RESULTS

In this section, we evaluate the proposed methods on two representative KG embedding tasks: entity alignment and KG completion. For each task, we conduct experiments on several commonly used real-world datasets, and report the results compared with several state-of-the-art methods.

4.1 Entity Alignment Results

4.1.1 Entity alignment datasets. We conduct the entity alignment experiments on the DBP15k [35] and DWY100k [36] datasets following the latest progress [3, 35–37]. DBP15k contains three datasets built from multi-lingual DBpedia: DBP_{ZH-EN} (Chinese to English), DBP_{JA-EN} (Japanese to English), and DBP_{FR-EN} (French to English). All the above datasets include 15,000 entity pairs as seed alignments. DWY100k consists of two large-scale cross-resource datasets: DBP-WD (DBpedia to Wikidata) and DBP-YG (DBpedia to YAGO3). Each dataset has 100,000 entity pairs as seed alignments. Following the previous work [3, 37], we split 30% of the entity seed alignments as training data, and leave the remaining data for the test. We list the statistics of the two datasets in Table 1.

4.1.2 Implementation Details. In the experiments, the hyperparameters of GCN (baseline) are taken from [3]. To make a fair comparison, we set the hyperparameters of RPE-GCN and the baseline model to the same. We set the embedding size to 128. We stack two layers (L=2) of GCN. We set $\gamma = 1.0$ and $\lambda = 0.5$. More details about the hyperparameters are listed in the Appendix D.

Table 2: Entity alignment results on DBP15k and DWY100k. Results of MTransE, JAPE, AlignE, GCN-Align, and MuGNN are taken from [3]. Other results are taken from [37]. The best results are in bold, and the second best results are underlined.

Methods	DBP _{ZH-EN}			DBP _{JA-EN}			DBP _{FR-EN}			DBP – WD			DBP – YG		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MTransE [5]	0.308	0.614	0.364	0.279	0.575	0.349	0.244	0.556	0.335	0.281	0.520	0.363	0.252	0.493	0.334
IPTransE [51]	0.406	0.735	0.516	0.367	0.693	0.474	0.333	0.685	0.451	0.349	0.638	0.447	0.297	0.558	0.386
JAPE [35]	0.412	0.745	0.490	0.363	0.685	0.476	0.324	0.667	0.430	0.318	0.589	0.411	0.236	0.484	0.320
AlignE [36]	0.472	0.792	0.581	0.448	0.789	0.563	0.481	0.824	0.599	0.566	0.827	0.655	0.633	0.848	0.707
GCN-Align [43]	0.413	0.744	0.549	0.399	0.745	0.546	0.373	0.745	0.532	0.506	0.772	0.600	0.597	0.838	0.682
SEA [29]	0.424	0.796	0.548	0.385	0.783	0.518	0.400	0.797	0.533	0.518	0.802	0.616	0.516	0.736	0.592
RSN [10]	0.508	0.745	0.591	0.507	0.737	0.590	0.516	0.768	0.605	0.607	0.793	0.673	0.689	0.878	0.756
MuGNN [3]	0.494	<u>0.844</u>	0.611	0.501	<u>0.857</u>	0.621	0.495	<u>0.870</u>	0.621	0.616	0.897	0.714	0.741	0.937	0.810
AliNet [37]	<u>0.539</u>	0.826	<u>0.628</u>	<u>0.549</u>	0.831	<u>0.645</u>	<u>0.552</u>	0.852	<u>0.657</u>	0.690	<u>0.908</u>	0.766	<u>0.786</u>	<u>0.943</u>	<u>0.841</u>
GCN [16]	0.477	0.828	0.593	0.495	0.848	0.613	0.498	0.861	0.619	0.611	0.891	0.707	0.747	0.939	0.814
RPE-GCN	0.576	0.878	0.678	0.582	0.883	0.685	0.597	0.899	0.701	<u>0.678</u>	0.915	<u>0.762</u>	0.797	0.957	0.854

Table 3: KG completion results on WN18RR, FB15k-237, and YAGO3-10. Results of TransE and RotatE are taken from [26] and [34], respectively. Other results are taken from [7]. The best results are in bold.

Methods	WN18RR				FB15k-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE [2]	0.226	-	-	0.501	0.294	-	-	0.465	-	-	-	-
DistMult [49]	0.43	0.39	0.44	0.49	0.241	0.155	0.263	0.419	0.34	0.24	0.38	0.54
ConvE [7]	0.43	0.40	0.44	0.52	0.325	0.237	0.356	0.501	0.44	0.35	0.49	0.62
ComplEx [39]	0.44	0.41	0.46	0.51	0.247	0.158	0.275	0.428	0.36	0.26	0.40	0.55
RotatE [34]	0.476	0.428	0.492	0.571	0.338	0.241	0.375	0.533	0.495	0.402	0.550	0.670
RPE-RotatE	0.501	0.457	0.515	0.585	0.355	0.261	0.390	0.546	0.546	0.460	0.598	0.703

By convention [3, 37, 43], we choose Mean Reciprocal Rank (MRR) and Hits at N (H@N) as the evaluation metrics. MRR is the average of the reciprocal of the rank results. H@N indicates the percentage of the targets that have been correctly ranked in top N. Higher MRR or H@N indicates better performance.

4.1.3 Main Results. In the experiments, we choose several state-of-the-art embedding-based entity alignment methods for comparison: MTransE [5], IPTransE [51], JAPE [35], AlignE [36], GCN-Align [43], SEA [29], RSN [10], MuGNN [3], AliNet [37], and the baseline GCN [16]. As claimed by [37], the recent GNN-based models like GMNN [48] and RDGCN [45] incorporate the surface information of entities into the embeddings. As our model mainly relies on structure and semantic information in the KG, we do not take these models into comparison following [37].

We present the entity alignment results in Table 2. It is evident that the proposed RPE-GCN achieves much better results than the baseline GCN. Specifically, on the DBP15k datasets, RPE-GCN gains about 10% higher H@1, 4% higher H@10, and 0.08 higher MRR than the baseline. Moreover, on the DWY100k datasets, RPE-GCN gains about 5% higher H@1, 2% higher H@10, and 0.05 higher MRR than the baseline. The results demonstrate the effectiveness of our proposed relational prototype entities.

Also, it is worth mentioning that the proposed RPE-GCN shows larger superiority on the DBP15k datasets against the recent AliNet

[37]. On the DBP15k datasets, RPE-GCN can achieve a gain of about 5% by H@1 and H@10, and 0.05 by MRR. However, on the DWY100k datasets, we find that RPE-GCN outperforms AliNet with an absolute margin of 0.01 by all the three metrics on DBP-YG, and obtains comparable performance to AliNet on DBP-WD. We know that the DWY100k datasets are extracted from different KGs, and the schema heterogeneity of them is much heavier than that of the DBP15k datasets which are extracted from the multi-lingual DBpedia [37]. We argue that the results may be caused by that AliNet is sophisticatedly designed for mitigating the non-isomorphism of neighborhood structures for entity alignment. Therefore, AliNet is particularly suitable for the DWY100k datasets. Recall that, our proposed RPE-GCN aims at exploiting global semantic similarities in the KG, and we leave the problem of the non-isomorphism of neighborhood structures in the KG for future work.

4.2 KG Completion Results

4.2.1 KG completion datasets. We evaluate our proposed models on the KG completion task on three commonly used datasets: WN18RR [38], FB15k-237 [7], and YAGO3-10 [24]. WN18RR, FB15k-237, and YAGO3-10 are subsets of WN18 [2], FB15k [2], and YAGO3 [24], respectively. As pointed out by [7, 38], WN18 and FB15k suffer

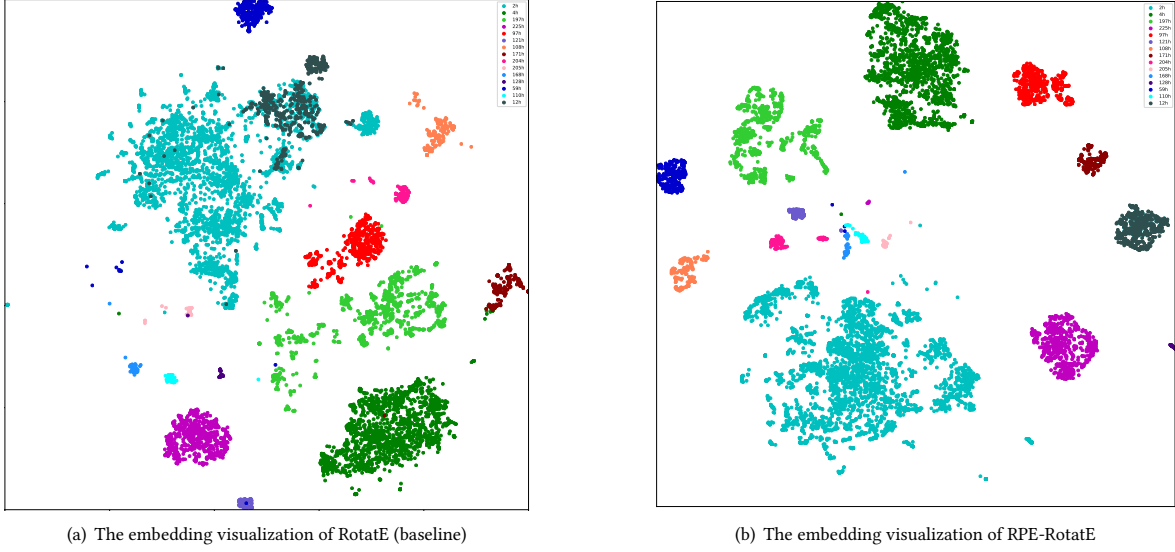


Figure 3: The embedding visualization of entities on the FB15k-237 dataset. Figure 3(a) and Figure 3(b) show the entity embedding visualization of baseline and our method, respectively. Different colors represent different categories.

Table 4: The Davies-Bouldin Index (DBI) performance of the baselines and our methods.

	DBP15k						DWY100k				WN18RR	FB15k-237	YAGO3-10
	ZH	EN	JA	EN	FR	EN	DBP	WD	DBP	YG			
Baseline	10.70	12.51	13.41	15.18	17.52	15.75	33.46	39.53	25.43	65.98	15.28	38.86	30.23
Our method	6.83	7.86	7.84	8.78	10.52	9.86	9.54	8.20	9.22	16.18	4.49	3.06	6.22

Table 5: Statistics of the number of relations, entities, and triples in each split for WN18RR, FB15k-237 and YAGO3-10.

Datates	#Relation	#Entity	#Training	#Validation	#Test
WN18RR	11	40,943	86,835	3,034	3,134
FB15k-237	237	14,541	272,115	17,535	20,466
YAGO3-10	37	123,182	1,079,040	5,000	5,000

from the test set leakage problem through inverse relations. Therefore, we use WN18RR and FB15k-237 as the benchmarks following [7, 34, 39]. The statistics of the datasets are summarized in Table 5.

4.2.2 Implementation Details. For comparing with the baseline RotatE fairly, the hyperparameters of RPE-RotatE are the same as those provided by the original paper [34]. Due to lack of space, we list the hyperparameters in the Appendix D.

Following [2, 34], we evaluate the performance of KG completion task in the filtered setting; that is, we rank test triples against all other candidate triples not appearing in the training, validation, or test set, where candidates are generated by corrupting head or tail entities: (h', r, t) or (h, r, t') . In the experiments, we also use MRR and H@N as the evaluation metrics.

4.2.3 Main Results. On the KG completion task, we compare the proposed method against several state-of-the-art methods, including TransE [2], DistMult [49], ComplEx [39], ConvE [7], and the baseline RotatE [34].

In Table 3, we show the performance of RPE-RotatE and several previous methods. We can see that the proposed RPE-RotatE significantly outperforms the baseline RotatE and achieves the best performance on all datasets, which demonstrates the effectiveness of our proposed method. Moreover, we can see that the improvement of RPE-RotatE on YAGO3-10 is much more significant than those on WN18RR and FB15k-237. Specifically, on the YAGO3-10 dataset, RPE-RotatE gains 0.051 higher MRR, 5.8% higher H@1, 4.8% higher H@3, and 3.3% higher H@10 than RotatE. We know that the relations in the YAGO3-10 dataset have clear semantic property, e.g., *directed* and *hasChild*; that is, the head entities of *directed* should be the “director”, and the tail entities of it should be the “film”. Therefore, we can expect that our proposed model is capable of working well on this dataset. WN18RR consists of relations such as *hypernym* and *similar to*, and such relations make the semantics of head and tail entities on this dataset less clear than YAGO3-10. FB15k-237 contains relations of composition patterns, which makes the relation types more complex than YAGO3-10. **However, on the WN18RR and FB15k-237 datasets, our proposed RPE-RotatE can**

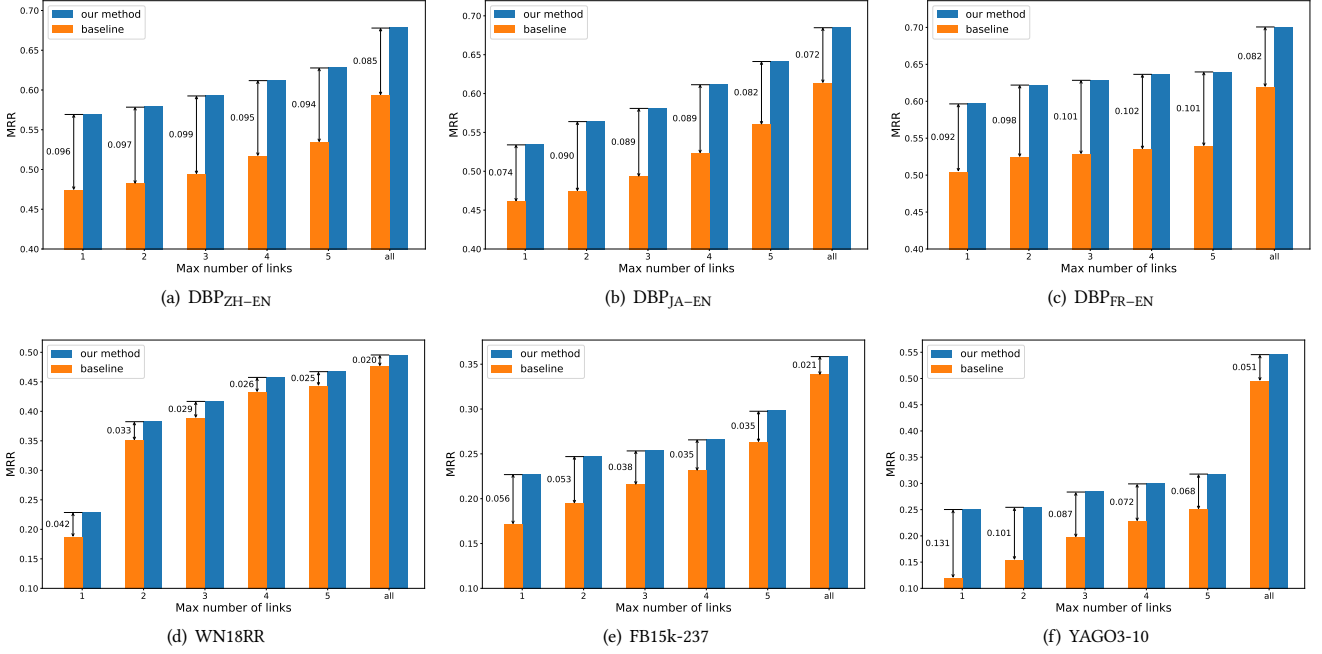


Figure 4: MRR results of entities w.r.t different number of links. If the max number of links is N , the entities have neighbors no larger than N . The general trend is that the smaller the max number of links, the more significant our model will improve the performance of the baseline.

also gain better performance than RotatE, which shows that our model is widely applicable.

5 FURTHER EXPERIMENTS

5.1 The Performance of Entity Clustering

We want to observe how our proposed method affects the embeddings of entities from the same and different categories. Figure 3 shows the embedding visualization of entities on the FB15k-237 dataset via t-SNE [23]. Notice that, an entity may belong to more than one category, e.g., there exists (h_1, r_1, t_1) and (h_1, r_2, t_2) ; that is, h_1 belongs to the head categories of r_1 and r_2 at the same time. For better visualization, we choose the categories with no overlap entities. Moreover, we only show the embeddings of head categories with more than 100 entities. Figure 3(a) shows the entity embeddings of the baseline, and Figure 3(b) shows the entity embeddings of our proposed method. Different colors represent different categories. It is readily apparent that the boundaries between different categories are more clear and the distances within categories are closer in our model, especially for the categories 110h and 12h (12h represents the head category of the relation with id 12). The visualization result demonstrates that our proposed method makes entity embeddings from different categories more distinguishable.

Moreover, we use the Davies-Bouldin Index (DBI) [6]—a metric for evaluating clustering algorithms—to measure the performance of entity clustering of the baselines and our methods. DBI is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. A lower DBI value means that the performance

of the clustering is better. For more details about DBI, please refer to [6]. We show the results in Table 4. We can see that our proposed methods achieve better performance than the baselines, which demonstrates that the proposed relational prototype entities can effectively encourage the semantic similarities of the entity embeddings from the same category.

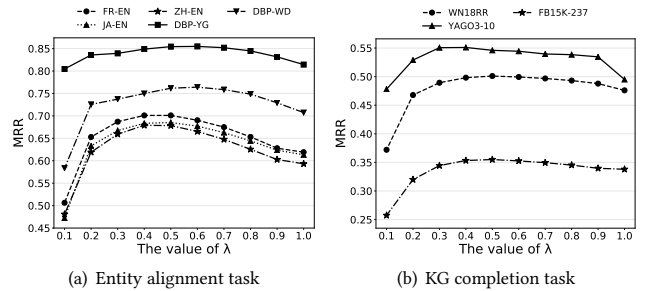


Figure 5: MRR results w.r.t the value of λ .

5.2 The Performance of Long-tail Entities

As discussed before, many existing methods focus on the local information in the KG. Under such a strategy, the long-tail entities that have few relational triples would only receive limited attention [10]. However, our proposed method can provide rich global semantic information for these entities through relational prototype entities.

Therefore, we also want to assess the effectiveness of the proposed methods on long-tail entities.

Figure 4 shows the MRR results of the baselines and our methods on long-tail entities w.r.t different number of links. Notice that, if the max number of links is N , it represents the entities with one-hop neighbors no larger than N . The ‘all’ represents the overall performance of the whole dataset. Due to lack of space, we only show the results of six datasets in Figure 4, and we show the results of the DBP-WD and DBP-YG datasets in the Appendix B. We can see that the general trend is that the smaller the max number of links, the more significant our model will improve the performance of the baselines. The results demonstrate the effectiveness of our methods, especially for entities with few relational triples.

5.3 Sensitivity to The Hyperparameter λ

In this section, we want to observe how the hyperparameter λ affects the performance of our method. In Figure 5, we show the MRR results on all the datasets w.r.t the value of λ . Recall that, when $\lambda = 1$, it represents the results of the baselines (GCN and RotatE). From the results, we can see that our proposed methods outperform the baselines when $0.3 \leq \lambda < 1$, which demonstrate the effectiveness of our approach. In this paper, we set $\lambda = 0.5$ on the entity alignment and KG embedding tasks.

6 CONCLUSION

In this paper, we exploit global semantic similarities in the KG by relational prototype entities. By enforcing the entities’ embeddings close to their associated prototypes’ embeddings, our approach can effectively encourage the global semantic similarities of entities connected by the same relation. Experiments on the entity alignment and KG completion tasks demonstrate that our proposed methods significantly improve the performance over the baselines, with overall performance better than recent state-of-the-art methods. Future work includes studying a unified architecture that can simultaneously exploit global semantic similarities and address structural heterogeneity in the knowledge graph.

REFERENCES

- [1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*.
- [2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NeurIPS*.
- [3] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In *ACL*.
- [4] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI*.
- [5] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*.
- [6] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. In *TPAMI*.
- [7] Tim Dettmers, Minervini Pasquale, Stenatorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*.
- [8] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shao-hua Sun, and Wei Zhang. 2014. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *SIGKDD*.
- [9] Jadagrad John Duchi, Elad Hazan, and Yoram Singer. [n. d.]. Adaptive subgradient methods for online learning and stochastic optimization. In *JMLR*.
- [10] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs. In *ICML*.
- [11] Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *IJCNLP*.
- [12] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. In *WSDM*.
- [13] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL*.
- [14] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*.
- [15] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- [16] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [17] Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *ISWC*.
- [18] 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. In *Semantic Web*.
- [19] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*.
- [20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*.
- [21] Hanxiao Liu, Yuxin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-relational Embeddings. In *ICML*.
- [22] Shiheng Ma, Jianhui Ding, Weijia Jia, Kun Wang, and Minyi Guo. 2017. Trans: Type-based multiple embedding representations for knowledge graph completion. In *ECML PKDD*.
- [23] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. In *JMLR*.
- [24] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.
- [25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*.
- [26] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *NAACL*.
- [27] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *AAAI*.
- [28] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-way Model for Collective Learning on Multi-relational Data. In *ICML*.
- [29] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-Supervised Entity Alignment via Knowledge Graph Embedding with Awareness of Degree Difference. In *WWW*.
- [30] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *ISWC*.
- [31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.
- [32] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In *NeurIPS*.
- [33] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.
- [34] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.
- [35] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*.
- [36] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *IJCAI*.
- [37] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge Graph Alignment Network with Gated Multi-hop Neighborhood Aggregation. In *AAAI*.
- [38] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *The Workshop on CVSC*.
- [39] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML*.
- [40] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *CIKM*.
- [41] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. In *TKDE*.
- [42] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *SIGKDD*.
- [43] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*.

- [44] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*.
- [45] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*.
- [46] Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Hierarchical Types.. In *IJCAI*.
- [47] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *WWW*.
- [48] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *ACL*.
- [49] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.
- [50] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI*.
- [51] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings.. In *IJCAI*.

A ANALYSIS ON ENTITY ALIGNMENT TASK

In this section, we analyze our proposed approach on the entity alignment task. For the entity alignment task, there exist two KGs, i.e., $\mathcal{G}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1)$ and $\mathcal{G}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2)$. Following the definitions in Section 3.3, we define the sets of relational prototype areas for \mathcal{G}_1 and \mathcal{G}_2 as C_1 and C_2 , respectively. We assume that for a relational prototype area C_{sr} in C_1 , its corresponding alignment relational prototype area C_{tg} in C_2 satisfies that $C_{sr} = C_{tg}$. Based on the loss function proposed by (7), the score function of entity alignment is

$$f(\mathbf{e}_{sr}, \mathbf{e}_{tg}) = -d(\mathbf{e}_{sr}, \mathbf{e}_{tg}) = -\|\mathbf{e}_{sr} - \mathbf{e}_{tg}\|, \mathbf{e}_{sr} \in \mathcal{E}_1, \mathbf{e}_{tg} \in \mathcal{E}_2. \quad (9)$$

Now, we want to prove that our approach can encourage the global semantic similarities of entities connected by the same relation on the entity alignment task. However, it's hard to prove this property theoretically on the entity alignment task. The main difficulties are twofold. First, the activation function in (4) makes the aggregation operation in RPE-GCN be non-linear. Second, the update of an entity embedding in RPE-GCN depends on the embeddings of its neighbors (including entities and relational prototype entities). However, even entities connected by the same relation may have different neighbors, which makes their updates of embeddings diverse. Therefore, such properties make the theoretical analysis—the embeddings of entities connected by the same relation will become closer to each other—of our approach for entity alignment task difficult.

However, based on the experiments in Section 5.1, a reasonable assumption is that the embeddings of entities connected by the same relation can become closer to each other on the entity alignment task. Then, we begin with this assumption and further get the following theorems.

THEOREM A.1. *For an entity $e \in \mathcal{E}_1$, there exists a relational prototype area $C_{sr} \in C_1$ such that $e \in C_{sr}$. Suppose that its corresponding alignment relational prototype area is $C_{tg} \in C_2$, and the radius of C_{tg} is R . If $d(C, C_{tg}) > 2R$ for any $C \in C_2 \setminus \{C_{tg}\}$, we have $f(e, \mathbf{e}_1) > f(e, \mathbf{e}_2)$ for any $\mathbf{e}_1 \in C_{tg}, \mathbf{e}_2 \notin C_{tg}, \mathbf{e}_1, \mathbf{e}_2 \in \mathcal{E}_2$.*

PROOF. Since $C_{sr} = C_{tg}$, for any $\mathbf{e} \in C_{sr}, \mathbf{e} \notin C_{tg}$, we have

$$d(\mathbf{e}_2, C_{tg}) > 2R \Rightarrow d(\mathbf{e}_2, C_{sr}) > 2R \Rightarrow d(\mathbf{e}_2, \mathbf{e}) > 2R.$$

For any $\mathbf{e}_1 \in C_{tg}$, we have $\mathbf{e}_1 \in C_{sr}$ since $C_{sr} = C_{tg}$. Therefore, we can get

$$d(\mathbf{e}_1, \mathbf{e}) \leq 2R < d(\mathbf{e}_2, \mathbf{e}).$$

Therefore, we can get $f(\mathbf{e}, \mathbf{e}_1) > f(\mathbf{e}, \mathbf{e}_2)$ based on (9), which completes the proof. \square

THEOREM A.2. *For an entity $e \in \mathcal{E}_2$, there exists a relational prototype area $C_{tg} \in C_2$ such that $e \in C_{tg}$. Suppose that its corresponding alignment relational prototype area is $C_{sr} \in C_1$, and the radius of C_{sr} is R . If $d(C, C_{sr}) > 2R$ for any $C \in C_1 \setminus \{C_{sr}\}$, we have $f(e, \mathbf{e}_1) > f(e, \mathbf{e}_2)$ for any $\mathbf{e}_1 \in C_{sr}, \mathbf{e}_2 \notin C_{sr}, \mathbf{e}_1, \mathbf{e}_2 \in \mathcal{E}_1$.*

PROOF. We omit the proof of Theorem A.2 since it is similar to the proof of Theorem A.1. \square

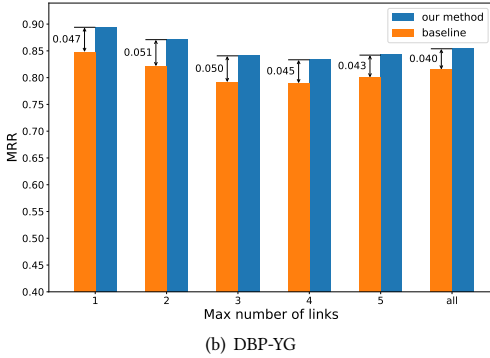
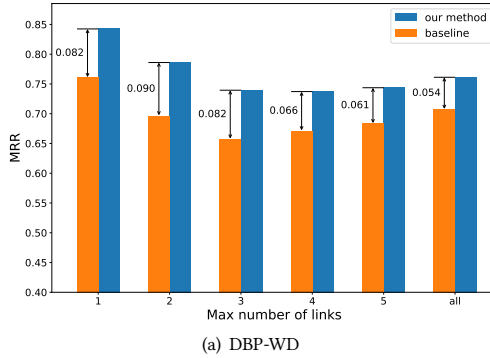
Form Theorem A.1 and Theorem A.2, we can see that the score of entities from the category of the correct answer will be higher than entities from other categories, which benefits the performance.

Table 6: Results of ablation study on the DBP15k and DWY100k datasets, where “w/o aggregation” represents “without aggregation” proposed by (6).

Methods	DBP _{ZH-EN}			DBP _{JA-EN}			DBP _{FR-EN}			DBP – WD			DBP – YG		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
GCN (w/o aggregation)	0.463	0.833	0.585	0.488	0.852	0.608	0.477	0.862	0.604	0.610	0.900	0.710	0.745	0.941	0.813
GCN	0.477	0.828	0.593	0.495	0.848	0.613	0.498	0.861	0.619	0.611	0.891	0.707	0.747	0.939	0.814
RPE-GCN (w/o aggregation)	0.554	0.884	0.665	0.561	0.889	0.673	0.573	0.906	0.688	0.663	0.923	0.754	0.789	0.958	0.849
RPE-GCN	0.576	0.878	0.678	0.582	0.883	0.685	0.597	0.899	0.701	0.678	0.915	0.762	0.797	0.957	0.854

B MORE RESULTS OF LONG-TAIL ENTITIES

Figure 6 shows the MRR results of long-tail entities on the DBP-WD and DBP-YG datasets. From the results, an interesting observation is that the MRR results of long-tail entities is better than others. However, the general trend is the same as the results in Section 5.2; that is, the smaller the max number of links, the more significant our model will improve the performance of the baselines. Recall that, if the max number of links is N , the entities have neighbors no larger than N .

**Figure 6: MRR results of entities w.r.t different number of links on DBP-WD and DBP-YG datasets.**

C ABLATION EXPERIMENTS

In Table 6, we show the ablation study of the aggregation strategy in (6). From the results, we find the strategy—which gets the entity embedding by aggregating the hidden states of all the layers—can

Table 7: Hyperparameters of RotatE and RPE-RotatE. k is the embedding size, b is the batch size, b' is the negative sampling size, γ is the fixed margin in the loss function, α is the temperature of negative sampling, and λ is the weight coefficient.

Dataset	k	b	b'	γ	α	λ	learning rate
WN18RR	500	512	1024	6.0	0.5	0.5	0.00005
FB15k-237	1000	1024	256	9.0	1.0	0.5	0.00005
YAGO3-10	500	1024	400	24.0	1.0	0.5	0.0002

Table 8: Hyperparameters of GCN and RPE-GCN. k is the embedding size, γ is the fixed margin in the loss function, L is the number of GCN layers, λ is the weight coefficient, and L_2 is the L2 regularizer weight.

Dataset	k	γ	L	λ	L_2	learning rate	dropout
DBP _{ZH-EN}	128	1.0	2	0.5	0.01	0.001	0.2
DBP _{JA-EN}	128	1.0	2	0.5	0.01	0.001	0.2
DBP _{FR-EN}	128	1.0	2	0.5	0.01	0.001	0.2
DBP – WD	128	1.0	2	0.5	0.01	0.001	0.2
DBP – YG	128	1.0	2	0.5	0.01	0.001	0.2

boost the performance on H@1 and MRR for both GCN and RPE-GCN models. It demonstrates the effectiveness of the used strategy. Also, we can see that the used strategy can only obtain comparable performance on H@10 to the strategy that only uses the hidden outputs at the last layer, which means that our used strategy mainly increase the percentage of the targets that have been correctly ranked first.

D HYPERPARAMETERS

We list the hyperparameters of our proposed methods on the entity alignment and KG completion tasks in Tables 8 and 7, respectively. To make a fair comparison, we set the hyperparameters of our models and the baseline models to the same. We use Adagrad [?] as the optimizer on the entity alignment task, and use Adam [15] as the optimizer on the KG completion task. Notice that, the best hyperparameter settings of GCN and RotatE are taken from MuGNN [3] and RotatE [34], respectively.