

Relation-Aware Neighborhood Matching Model for Entity Alignment

Yao Zhu¹, Hongzhi Liu^{2*}, Zhonghai Wu^{3,4}, Yingpeng Du²

¹ Center for Data Science, Peking University, Beijing, China

² School of Software and Microelectronics, Peking University, Beijing, China

³ National Engineering Center of Software Engineering, Peking University, Beijing, China

⁴ Key Lab of High Confidence Software Technologies (MOE), Peking University, Beijing, China
{yao.zhu, liuhz, wuzh, dyp1993}@pku.edu.cn

Abstract

Entity alignment which aims at linking entities with the same meaning from different knowledge graphs (KGs) is a vital step for knowledge fusion. Existing research focused on learning embeddings of entities by utilizing structural information of KGs for entity alignment. These methods can aggregate information from neighboring nodes but may also bring noise from neighbors. Most recently, several researchers attempted to compare neighboring nodes in pairs to enhance the entity alignment. However, they ignored the relations between entities which are also important for neighborhood matching. In addition, existing methods paid less attention to the positive interactions between the entity alignment and the relation alignment. To deal with these issues, we propose a novel Relation-aware Neighborhood Matching model named RNM for entity alignment. Specifically, we propose to utilize the neighborhood matching to enhance the entity alignment. Besides comparing neighbor nodes when matching neighborhood, we also try to explore useful information from the connected relations. Moreover, an iterative framework is designed to leverage the positive interactions between the entity alignment and the relation alignment in a semi-supervised manner. Experimental results on three real-world datasets demonstrate that the proposed model RNM performs better than state-of-the-art methods.

Introduction

In knowledge graphs (KGs), facts are presented as triples of (h, r, t) , indicating there is a relation r from the head entity h to the tail entity t . Real-world KGs such as DBpedia (Lehmann et al. 2015), YAGO (Suchanek, Kasneci, and Weikum 2007), and Freebase (Bollacker et al. 2008), which store a great deal of knowledge, have been employed in various applications like recommendation systems (Cao et al. 2019b), question answering (Huang et al. 2019), and search engines (Xiong, Power, and Callan 2017).

However, each individual KG may be incomplete. Since different KGs are constructed independently from different data sources, they are usually complementary to each other. Therefore, integrating heterogeneous knowledge from various KGs has become an urgent issue. Entity alignment is a vital step for knowledge fusion from different KGs, which

aims at linking entities with the equivalent meaning from different KGs. The facts can consequently be fused based on the aligned entities.

Regarding the entity alignment task, most of the existing research focused on constructing embedding-based models. These methods tried to embed the entities of KGs into a latent space and calculated the distances between entity vectors as the evidences for alignment. TransE (Bordes et al. 2013), as an effective KG embedding model, has been widely adopted for entity alignment (Hao et al. 2016; Chen et al. 2017; Zhu et al. 2017; Sun et al. 2018). To better utilize the information from neighbors, graph convolutional networks (GCNs) (Kipf and Welling 2017) were utilized to improve the representation learning of entities (Wang et al. 2018; Wu et al. 2019a; Ye et al. 2019; Sun et al. 2020). However, these methods concentrated on learning comprehensive embeddings for entities, meanwhile, may bring additional noise from neighbors.

Recently, several studies tried to conduct subgraph matching when comparing the candidate entity pairs to enhance the alignment (Xu et al. 2019; Wu et al. 2020). However, these methods only compared the neighboring entities but ignored the connected relations which also contain important information for neighborhood matching and entity alignment. Moreover, existing methods paid less attention to the positive interactions between the entity alignment task and the relation alignment task. Our insights are described as follows. First, neighborhood matching with relations can enhance the reliability of entity alignment. Figure 1 shows an example of the entity alignment with neighborhood matching. Assume the entities *Rome*, *Renaissance*, *Florence* and *Michelangelo* in the two KGs have been aligned. If we only consider the neighboring entities when matching subgraphs, the entity *Italy* (in Chinese) in KG1 is more likely to be misaligned with the entity *David Statue* in KG2. However, if we compare the connected relations at the same time and consider the 1-to-1 property of relation *capital*, the entity *Italy* can be correctly aligned crossing two KGs. This implies that relations play a significant role in neighborhood matching not only for the semantic meaning but also for the mapping property. Second, relation alignments can help to find the alignments of entities, and on the other hand, entity alignments can also assist the relation alignment task. Specifically, the entity alignment can be inferred based on

*Corresponding author

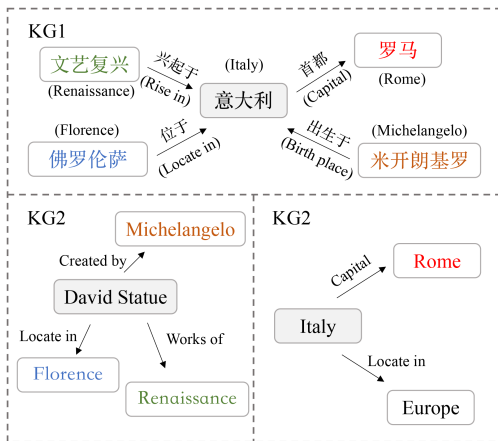


Figure 1: An example to illustrate the importance of relations when matching neighborhood. The upper part of the figure is a subgraph of KG1 (in Chinese), while the lower parts are two subgraphs of KG2 (in English). Assume the entities *Rome*, *Renaissance*, *Florence* and *Michelangelo* in the two KGs have been aligned. The entity *Italy* in KG1 is more likely to be misaligned with the entity *David Statue* in KG2 with consideration of the neighbor entities while ignoring the connected relations. However, the two *Italy* entities in different KGs can be correctly aligned by matching both the relation (Capital) and the neighbor (Rome).

the neighboring entities and the linking relations, while the relation alignment can be inferred from the connected head and tail entities. Thus, it is reasonable to implement both entity alignment and relation alignment in a unified framework.

Therefore, in this paper, we propose a novel Relation-aware Neighborhood Matching model named RNM for entity alignment. Besides comparing neighboring entities when matching subgraphs, we also exploit the semantic information and mapping properties from linking relations for entity alignment. The semantic information of relations helps us with the relation matching in neighborhood, while mapping properties of relations provide the probability of alignment. Moreover, we design an iterative framework to unify the entity alignment and the relation alignment, in which the two tasks can reinforce each other in a semi-supervised manner. Experimental results on three real-world datasets show that RNM significantly outperforms several state-of-the-art methods.

The remainder of this paper is organized as follows. First, we discuss the related work and introduce the problem definition in the following two sections. Then, we describe the proposed model in detail. After that, experimental settings and empirical evaluation results are presented. Finally, we conclude the paper and point out some future work.

Related Work

Most of the existing entity alignment methods focused on embedding entities from different KGs into the same la-

tent space and measured the alignment by calculating the distance between entity embeddings. TransE (Bordes et al. 2013), as one of the most practical models for KG embedding, has been adopted for entity alignment. MTransE (Chen et al. 2017) utilized TransE model to learn entity embeddings for two KGs separately and designed a space transformation mechanism for the alignment. Instead of training embeddings separately for different KGs, IPTransE (Zhu et al. 2017) employed a path-based TransE model to train the joint knowledge embeddings and proposed an iterative strategy to expand seed alignments. After that, for better iteration, BootEA (Sun et al. 2018) designed a bootstrapping alignment model based on translational embedding learning, and used constraints to reduce the error accumulation when iterating.

Since graph convolutional networks (GCNs) (Kipf and Welling 2017) have achieved remarkable progress in graph learning, some work tried to apply GCNs to entity alignment for better representation learning. Wang et al. (2018) proposed a GCN-Align model for entity alignment which trained GCNs to embed entities of each KG into a unified vector space. After that, relations were taken into account for entity alignment. HGCN (Wu et al. 2019b) jointly learned both entity and relation representations via a GCN-based framework and RDGCN (Wu et al. 2019a) constructed a dual relation graph for embedding learning. Moreover, AliNet (Sun et al. 2020) improved GCNs by aggregating multi-hop neighborhood with gated strategy and attention mechanism. These methods tried to make use of the structural and neighborhood information to learn better representations of entities. However, they may also bring in some noise from neighbors, which could degrade the performance of alignment.

More recently, some researchers attempted to employ subgraph matching for better entity alignments. Xu et al. (2019) formulated the KG-alignment task as a graph matching problem by introducing a local sub-graph for each entity. NMN (Wu et al. 2020) was a cross-graph neighborhood matching model which jointly encoded the difference of neighborhood for entity pairs. However, these methods only took the neighboring entities for comparison but ignored the connected relations which are also important for subgraph matching. Thus, in this paper, we propose a novel relation-aware neighborhood matching model which explores the semantic information and mapping properties of relations when conducting subgraph matching. Moreover, entities and relations are iteratively aligned in our model to make these two tasks reinforce each other.

Problem Definition

The entity alignment and the relation alignment are two related tasks for knowledge fusion.

Formally, a KG can be denoted as $G = (E, R, T)$, where E , R , and T are the sets of entities, relations, and triples, respectively. Given two heterogeneous KGs to be fused, which are $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$, we assume there is a set of pre-aligned entity pairs between the two KGs, which is defined as $\mathbb{L} = \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2, e_1 \text{ equals to } e_2\}$.

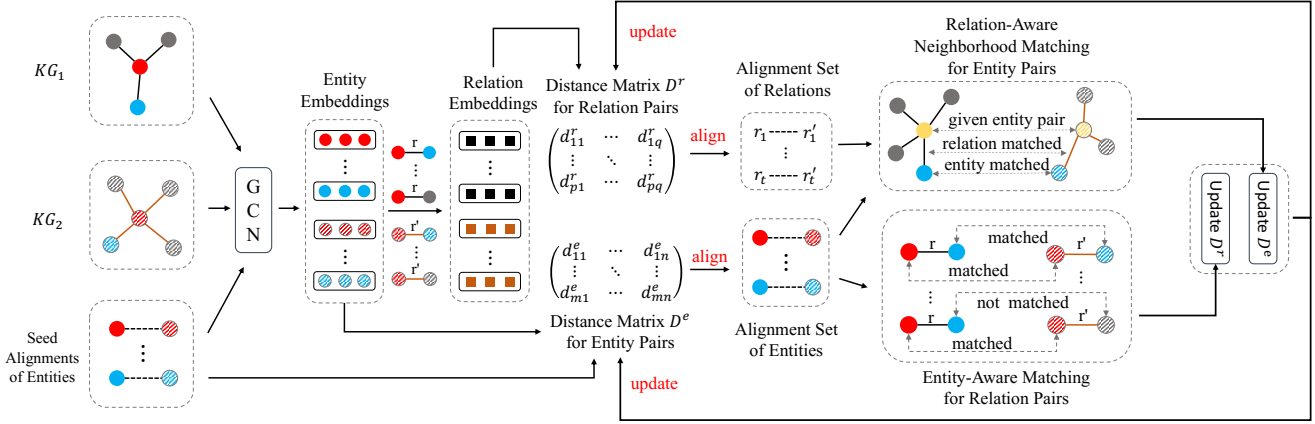


Figure 2: Overall architecture of the proposed model RNM. Assume that there are m entities in KG_1 , n entities in KG_2 , p relations in KG_1 , and q relations in KG_2 .

For the entity alignment task, our goal is to find out the remaining equivalent entity pairs. For the relation alignment task, our goal is to find out the relation pairs with the same meaning from the two given KGs. Note that the relation alignment is an unsupervised task in this paper.

Proposed Model

In this section, we will first give an overview of the proposed model RNM. After that, components of the model will be described in detail, which are embedding learning for entities and relations, relation-aware neighborhood matching for entity pairs, and entity-aware matching for relation pairs. Finally, we will present the iterative strategy and some implementation details of RNM.

Overview of RNM

Figure 2 illustrates the overall architecture of the proposed model RNM. First, given two KGs and a set of seed alignments of entities, we jointly learn the embeddings of entities and relations using GCNs with a TransE-like regularizer. After that, we iteratively align the entities and relations in a semi-supervised manner. In each iteration, we utilize the graph structure information to determine new matching pairs of entities and relations by a relation-aware neighborhood matching module and an entity-aware entity matching module, respectively.

Embedding Learning for Entity and Relation

To align the entities of two KGs, we embed them into the same latent space to make them comparable. Similarly, we embed the relations of the two KGs into the same latent space for relation alignment. To explore the interactions between entities and relations in the KG, we propose to jointly learn the embeddings of entities and relations.

Entity Embedding Given two KGs and a set of seed alignments of entities, we utilize GCNs to embed all the entities of the two KGs into the same latent space with con-

sideration of the structure information of the two KGs. Following (Xu et al. 2019; Wu et al. 2020), we initialize the entity representations with the pre-trained word embeddings which can provide useful semantic information of entities. Moreover, we adopt the highway strategy (Srivastava, Greff, and Schmidhuber 2015) to control the noise in the propagation procedure of GCNs with multiple layers.

We take the outputs of GCN stated above as the embeddings of entities, and define the final representations of all entities as $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_n | \tilde{\mathbf{x}}_i \in \mathbb{R}^{\tilde{d}}\}$, where \tilde{d} denotes the dimension of entity embeddings and n denotes the number of entities. For an entity pair (e_i, e'_j) where $e_i \in E_1$ and $e'_j \in E_2$, we define the distance between them as

$$d(e_i, e'_j) = \|\tilde{\mathbf{x}}_{e_i} - \tilde{\mathbf{x}}_{e'_j}\|_1, \quad (1)$$

where $\|\cdot\|_1$ denotes the 1-norm measure for vectors. Smaller $d(e_i, e'_j)$ indicates the higher probability of alignment between the two entities e_i and e'_j .

To embed the entities of two KGs into the same latent space, we take the seed alignments as training data and design a margin-based loss function for entity alignment as follows,

$$L_E = \sum_{(p,q) \in \mathbb{L}} \sum_{(p',q') \in \mathbb{L}'} \max\{0, d(p, q) - d(p', q') + \gamma\}, \quad (2)$$

where \mathbb{L} denotes the set of pre-aligned entity pairs, \mathbb{L}' is a set of negative alignments upon nearest neighbor sampling (Wu et al. 2020), and $\gamma > 0$ denotes the margin. The loss function assumes that the distance between aligned entity pairs should be close to zero, while the distance between negative samples should be as far as possible.

Relation Embedding In the KG, facts are encoded as triples, i.e., (h, r, t) , where h denotes the head entity, t denotes the tail entity, and r denotes the relation from h to t . Therefore, the meaning of a relation is associated with its two connected entities. To leverage the information of connected entities, we utilize the embeddings of head entities

and tail entities learned from GCNs to represent relations in the KG, which can be written as follows,

$$\mathbf{r} = \text{concat}[\mathbf{g}_r^h, \mathbf{g}_r^t], \quad (3)$$

where $\mathbf{r} \in \mathbb{R}^{2\bar{d}}$ denotes the embedding of the relation $r \in R_1 \cup R_2$, concat means the operation of concatenation, and \mathbf{g}_r^h and \mathbf{g}_r^t denote the average embeddings of all distinct head entities and tail entities for r , respectively.

Moreover, to further explore the translational information for relations based on triples, inspired by TransE (Bordes et al. 2013), we design a regularizer as follows,

$$\Omega_R = \sum_{(h,r,t) \in T_1 \cup T_2} \|\mathbf{h} + \mathbf{W}_R \mathbf{r} - \mathbf{t}\|_1, \quad (4)$$

where T_1 and T_2 denote the sets of triples for two given KGs G_1 and G_2 , respectively. $\mathbf{W}_R \in \mathbb{R}^{\bar{d} \times 2\bar{d}}$ denotes the transformation matrix from the latent relation space to latent entity space, which is the model parameter to be learned.

Objective function To jointly learn the embeddings of entities and relations, we formulate the objective function as follows,

$$L = L_E + \lambda \cdot \Omega_R, \quad (5)$$

where λ is a trade-off coefficient to balance the loss of entity alignment and the loss of regularizer with consideration of the embeddings of relations. Our goal is to minimize the function above after the pre-training of entity embeddings. In addition, we utilize Adam (Kingma and Ba 2015) for the objective optimization.

Relation-Aware Neighborhood Matching

GCNs aim to aggregate information from neighboring nodes but may also bring some additional noise from neighbors. To reduce the impact of these noise, we propose a relation-aware neighborhood matching model to compare entity pairs. We assume that if two entities from different KGs have been aligned, then with the relation of the same meaning, the alignment probability of two pointing tail entities can be inferred according to the mapping property of the relation. For instance, 1-to-1 relation can provide the exact alignment while 1-to-N relation can only show the probability of $1/N$.

For each candidate entity pair (e_i, e'_j) where $e_i \in G_1$ and $e'_j \in G_2$, besides comparing their one-hop neighbor entities in pairs, we also consider the comparison between connected relations. Specifically, let \mathcal{N}_{e_i} be the set of one-step neighbor entities of e_i in G_1 , and $\mathcal{N}_{e'_j}$ be the set of one-step neighbor entities of e'_j in G_2 . For neighborhood matching with respect to e_i and e'_j , we compare all the entity pairs and the connected relation pairs in $C_{ij}^e = \{(n_1, n_2), (r_1, r_2) | n_1 \in \mathcal{N}_{e_i}, n_2 \in \mathcal{N}_{e'_j}, (e_i, r_1, n_1) \in T_1, (e'_j, r_2, n_2) \in T_2\}$, where T_1 and T_2 are the sets of triples for the two KGs, respectively. After that, we focus on the matched neighbors with matched relations which are vital for entity alignment. Thus, the matched set M_{ij}^e is defined as the subset of C_{ij}^e , in which the elements satisfy $(n_1, n_2) \in \mathbb{L}_e$ and $(r_1, r_2) \in \mathbb{L}_r$, where \mathbb{L}_e denotes the alignment set of entities and \mathbb{L}_r denotes the alignment set of relations.

Moreover, mapping properties of connected relations are also important for entity alignment. Thus, for each matched case in M_{ij}^e , we will compute the alignment probability based on r_1, r_2 and n_1, n_2 , which can be written as follows,

$$P(r_1, r_2, n_1, n_2) = P(r_1, n_1) \cdot P(r_2, n_2) \quad (6)$$

where

$$P(r_1, n_1) = \frac{1}{|\{e | (e, r_1, n_1) \in T_1\}|} \quad (7)$$

and

$$P(r_2, n_2) = \frac{1}{|\{e | (e, r_2, n_2) \in T_2\}|}. \quad (8)$$

$P(r_1, n_1)$ and $P(r_2, n_2)$ denote the mapping probability with respect to the corresponding relation and neighbor entity, respectively. Thus, we can update the distance between two entities as follows,

$$d_{ij}^e = \|\tilde{\mathbf{x}}_{e_i} - \tilde{\mathbf{x}}_{e'_j}\|_1 - \lambda_e \cdot \frac{\sum_{M_{ij}^e} P(r_1, r_2, n_1, n_2)}{|\mathcal{N}_{e_i}| + |\mathcal{N}_{e'_j}|}, \quad (9)$$

where λ_e is a hyper-parameter to control the tradeoff between the embedding distance and the matching score. Greater matching score indicates the higher probability of alignment for the candidate entity pair.

Entity-Aware Relation Matching

For two relations from different KGs, we assume that the more alignments of head entities and tail entities are at the same time in their associated triples, the more likely the two relations are with the same meaning. For a relation r , we define $S_r = \{(h, t) | (h, r, t) \in T\}$ as the set of its related entity pairs, where T denotes the set of triples in the given KG. Thus, given a candidate relation pair (r_i, r'_j) where r_i from G_1 and r'_j from G_2 , we first form the corresponding entity pair sets S_{r_i} and $S_{r'_j}$. Then, we compare all entity pairs in $C_{ij}^r = \{(h_1, h_2), (t_1, t_2) | (h_1, t_1) \in S_{r_i}, (h_2, t_2) \in S_{r'_j}\}$ and define the matching set M_{ij}^r as the subset of C_{ij}^r where elements meet the conditions of $(h_1, h_2) \in \mathbb{L}_e$ and $(t_1, t_2) \in \mathbb{L}_e$. Therefore, the distance between the relation pair (r_i, r'_j) can be updated as follows,

$$d_{ij}^r = \|\mathbf{r}_i - \mathbf{r}'_j\|_1 - \lambda_r \cdot \frac{|M_{ij}^r|}{|S_{r_i}| + |S_{r'_j}|}, \quad (10)$$

where λ_r is a tradeoff coefficient. Similar as the distance measure for the entity pairs, we consider both the embedding distance and the matching score for relation pairs.

Iterative Strategy and Implementation Details

To make use of the positive interactions between the entity alignment task and the relation alignment task, we design a semi-supervised framework in which the entity alignment and the relation alignment can enhance each other iteratively. Let $D^e \in \mathbb{R}^{|E_1| \times |E_2|}$ denote the distance matrix for entity pairs from KG_1 to KG_2 , and $D^r \in \mathbb{R}^{|R_1| \times |R_2|}$ denote the distance matrix for relation pairs from KG_1 to KG_2 . Algorithm 1 presents the iterative strategy of RNM.

Algorithm 1 Iterative Strategy of RNM

Input: Entity embeddings $\{\tilde{\mathbf{x}}_i\}$, relation embeddings $\{\mathbf{r}_i\}$, seed alignments of entities \mathbb{L} , maximum number of iterations T .

Output: D^e (distance matrix of entities) and D^r (distance matrix of relations).

- 1: Initialize D^e using Eq. (11);
 - 2: Initialize D^r using Eq. (12);
 - 3: **repeat**
 - 4: Update alignment sets according to Algorithm 2;
 - 5: Update D^e using Eq. (9) with the consideration of relation-aware neighborhood matching;
 - 6: Update D^r using Eq. (10) with the consideration of entity-aware matching;
 - 7: **until** D^e, D^r are converged or the iteration reaches T
 - 8: **return** D^e, D^r
-

Algorithm 2 Update Alignment Sets

Input: D^e (distance matrix of entities), D^r (distance matrix of relations), distance threshold δ_e and δ_r .

Output: \mathbb{L}_e (alignment set of entities), \mathbb{L}_r (alignment set of relations).

- 1: Initialize $\mathbb{L}_e \leftarrow \emptyset, \mathbb{L}_r \leftarrow \emptyset$;
 - 2: **for** each entity e_i in KG_1 **do**
 - 3: $j = \arg \min_j d_{ij}^e$ // find the nearest entity in KG_2
 - 4: **if** $d_{ij}^e < \delta_e$ **then**
 - 5: $\mathbb{L}_e \leftarrow \mathbb{L}_e \cup (e_i, e'_j)$
 - 6: **end if**
 - 7: **end for**
 - 8: **for** each relation r_i in KG_1 **do**
 - 9: $j = \arg \min_j d_{ij}^r$ // find the nearest relation in KG_2
 - 10: **if** $d_{ij}^r < \delta_r$ **then**
 - 11: $\mathbb{L}_r \leftarrow \mathbb{L}_r \cup (r_i, r'_j)$
 - 12: **end if**
 - 13: **end for**
 - 14: For the conflicts in \mathbb{L}_e or \mathbb{L}_r , we will choose the pair with smaller distance;
 - 15: **return** $\mathbb{L}_e, \mathbb{L}_r$
-

The initialization of D^e is defined as follows with the learned embeddings of entities,

$$d_{ij}^e = \begin{cases} 0 & (e_i, e'_j) \in \mathbb{L} \\ \infty & (e_i, e'_k) \in \mathbb{L} \wedge j \neq k, \\ \|\tilde{\mathbf{x}}_{e_i} - \tilde{\mathbf{x}}_{e'_j}\|_1 & otherwise \end{cases}, \quad (11)$$

and the initialization of D^r can be written as follows with the learned embeddings of relations,

$$d_{ij}^r = \|\mathbf{r}_i - \mathbf{r}'_j\|_1. \quad (12)$$

D^e and D^r can be utilized for alignment ranking or alignment set generation. The method for generating or updating the alignment sets is shown in Algorithm 2.

In addition, we introduce the reverse relations to enrich the KGs. For instance, for the fact (Tokyo, *CapitalOf*, Japan), we will also build another triple (Japan, *Cap-*

Datasets		Ent.	Rel.	Tri.
<i>DBP15K_{ZH-EN}</i>	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
<i>DBP15K_{JA-EN}</i>	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
<i>DBP15K_{FR-EN}</i>	French	66,858	1,379	192,191
	English	105,889	2,209	278,590

Table 1: Statistics of datasets

italOf⁻¹, Tokyo). Thus, the set of relations and the set of triples of a given KG will be accordingly enlarged.

Experiments

Experimental Setup

Datasets To evaluate the performance of the proposed model, we utilize three cross-lingual datasets from DBP15K as the experimental data. These datasets are subsets of the large-scale knowledge graph DBpedia (Lehmann et al. 2015) and are selected from different language versions including English, Chinese, Japanese, and French. Each dataset consists of two KGs of different languages and 15,000 aligned entity pairs. Recently, these three datasets have been widely employed by researchers for entity alignment (Wu et al. 2019a; Sun et al. 2020; Wu et al. 2020). The statistic details of the datasets are shown in Table 1.

Experimental Settings We employ a 2-layer GCN to learn the entity embeddings. The dimension of hidden layer in GCN is set as 300. The learning rate is set to 0.001. Following (Wu et al. 2020), we first translate non-English entity names into English and then initialize the entity embeddings with the pre-trained word vectors from Glove model, and the proportion of seed alignments is set as 30%. Besides, we set the margin γ as 1, threshold δ_e as 5, threshold δ_r as 3, λ as 0.001, λ_e as 10, and λ_r as 200. We select the nearest 100 entities and the nearest 20 relations as candidates for matching. The number of negative samples for each positive one is set as 125, the maximum number of iterations T is set as 4. We first optimize Eq. (2) for 50 epochs, and then jointly train the embeddings using Eq. (5) for 10 epochs.

We utilize TensorFlow to implement the proposed model RNM. The experiments are conducted on a server with two Intel(R) Xeon(R) CPUs E5-2660 @ 2.20GHz, an NVIDIA Tesla P100 GPU and 16 GB memory.

Evaluation Metrics and Baselines The same as in previous work (Sun et al. 2018; Yang et al. 2020), we adopt Hits@ k and mean reciprocal rank (MRR) as the evaluation metrics. Hits@ k measures the proportion of correctly aligned entities ranked in the top k list. k is set as 1 and 10 as in previous work. MRR is calculated as the average of the reciprocal ranks of the results. Higher Hits@ k or MRR indicates the better performance of the model.

For comparison, we choose several competitive entity alignment methods as baselines and classify them into three categories: (1) TransE-based models, including MTransE (Chen et al. 2017), IPTransE (Zhu et al. 2017), BootEA (Sun et al. 2018), AKE (Lin et al. 2019), and SEA (Pei

Models	ZH-EN			JA-EN			FR-EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE (Chen et al. 2017)	30.8	61.4	0.364	27.9	57.5	0.349	24.4	55.6	0.335
IPTransE (Zhu et al. 2017)	40.6	73.5	0.516	36.7	69.3	0.474	33.3	68.5	0.451
BootEA (Sun et al. 2018)	62.9	84.8	0.703	62.2	85.4	0.701	65.3	87.4	0.731
AKE (Lin et al. 2019)	32.5	70.3	0.449	25.9	66.3	0.390	28.7	68.1	0.416
SEA (Pei et al. 2019)	42.4	79.6	0.548	38.5	78.3	0.518	40.0	79.7	0.533
GCN-Align (Wang et al. 2018)	41.3	74.4	0.549	39.9	74.5	0.546	37.3	74.5	0.532
KECG (Li et al. 2019)	47.8	83.5	0.598	49.0	84.4	0.610	48.6	85.1	0.610
MuGNN (Cao et al. 2019a)	49.4	84.4	0.611	50.1	85.7	0.621	49.5	87.0	0.621
NAEA (Zhu et al. 2019)	65.0	86.7	0.720	64.1	87.3	0.718	67.3	89.4	0.752
AliNet (Sun et al. 2020)	53.9	82.6	0.628	54.9	83.1	0.645	55.2	85.2	0.657
GMNN (Xu et al. 2019)	67.9	78.5	0.694	74.0	87.2	0.789	89.4	95.2	0.913
RDGCN (Wu et al. 2019a)	70.8	84.6	0.746	76.7	89.5	0.812	88.6	95.7	0.911
HGCN (Wu et al. 2019b)	72.0	85.7	0.768	76.6	89.7	0.813	89.2	96.1	0.917
NMN (Wu et al. 2020)	<u>73.3</u>	<u>86.9</u>	<u>0.781</u>	<u>78.5</u>	<u>91.2</u>	<u>0.827</u>	<u>90.2</u>	<u>96.7</u>	<u>0.924</u>
RNM	84.0	91.9	0.870	87.2	94.4	0.899	93.8	98.1	0.954

Table 2: Performance of different entity alignment methods

Models	ZH-EN			JA-EN			FR-EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
RNM	84.0	91.9	0.870	87.2	94.4	0.899	93.8	98.1	0.954
RNM (-AP)	81.8	91.6	0.856	85.7	94.4	0.890	93.0	98.0	0.945
RNM (-IS)	81.6	91.1	0.852	84.6	93.7	0.881	92.5	97.7	0.945
RNM (-RM)	78.5	90.6	0.830	83.3	93.6	0.871	91.3	97.1	0.935

Table 3: Ablation study of the proposed model

et al. 2019); (2) GCN-based models which only utilize structural information, including GCN-Align (Wang et al. 2018), KECG (Li et al. 2019), MuGNN (Cao et al. 2019a), NAEA (Zhu et al. 2019), and AliNet (Sun et al. 2020); (3) GCN-based models which employ entity name initialization, including GMNN (Xu et al. 2019), RDGCN (Wu et al. 2019a), HGCN (Wu et al. 2019b), and NMN (Wu et al. 2020). Note that GMNN and NMN are models with subgraph matching.

Experimental Results

Entity Alignment Table 2 shows the performance of different methods on the entity alignment task. The results of Hits@1 and Hits@10 are in percentage (%). Numbers in **bold** denote the best results among all models and the underlined ones denote the second best results. The experimental results show that RNM significantly outperforms all baselines on the three datasets. And it can achieve that all the values of Hits@1 higher than 80%, those of Hits@10 higher than 90%, and those of MRR higher than 0.85. It is worth noting that Hits@1 directly reflects the accuracy of alignment. Thus, the outstanding results in Hits@1 further confirms the effectiveness of the proposed model RNM.

Specifically, among all TransE-based models, BootEA performs the best because it adopts a bootstrap strategy to iteratively expand the seed alignments. This indicates that the iterative strategy can significantly improve the performance of entity alignment. And for GCN-based models that only consider the structural information, NAEA outperforms the others probably because it considers both neighboring nodes and relations when representing entities. This confirms that the relation information is important for entity alignment.

Moreover, NMN performs the best among all baselines. The improvements may come from its neighborhood matching module. However, the proposed model RNM further outperforms NMN by 10.7%, 8.7%, 3.6% in Hits@1. This confirms that neighborhood matching with relations can effectively improve the performance of entity alignment.

Ablation Study To evaluate the effectiveness of our designed modules, we construct several ablation studies on the proposed model RNM and Table 3 shows the results. Specifically, (1) **RNM (-AP)** denotes the model RNM without considering the alignment probability (Eq. (6)) in the relation-aware neighborhood matching module; (2) **RNM (-IS)** denotes RNM without the iterative strategy; (3) **RNM (-RM)** denotes RNM without considering relations when matching neighborhood for entity pairs. From the experimental results, we can observe that the performance of RNM (-RM) drops the most, which confirms that it is important to take into account the connected relations when matching neighborhood for entity pairs. It is noted that RNM (-IS) consistently outperforms all baselines in Table 2, which confirms the effectiveness of proposed relation-aware neighborhood matching module. In addition, RNM (-AP) is 2.2%, 1.5% and 0.8% lower than RNM in Hits@1 on three datasets, which indicates that considering mapping properties of relations could further improve the accuracy of entity alignment.

Relation Alignment The proposed model RNM can not only be used for entity alignment but also for relation alignment. Table 4 shows the comparison results of different methods on the three datasets for relation alignment. Note that the results of baselines are from the reported data in

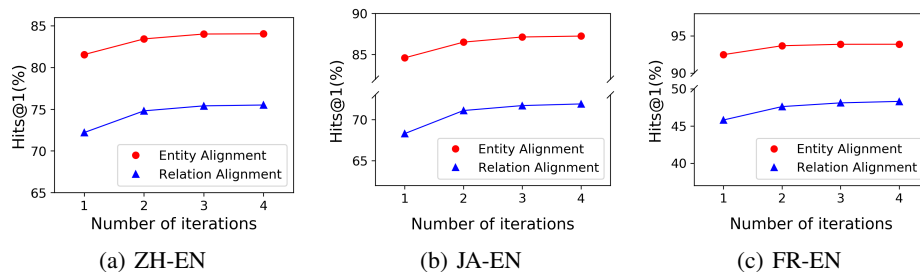


Figure 3: Results of entity alignment and relation alignment w.r.t the number of iterations.

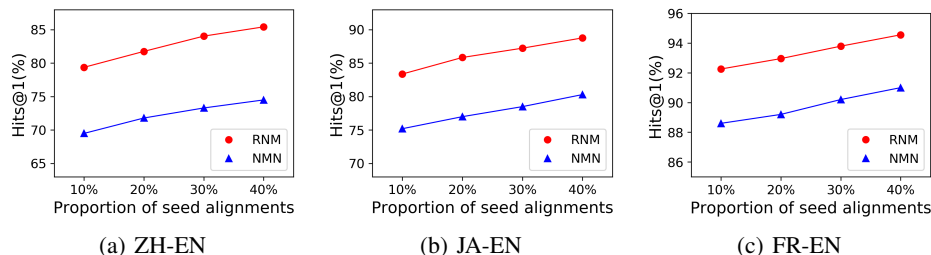


Figure 4: Results of entity alignment w.r.t the proportion of seed alignments.

Models	ZH-EN		JA-EN		FR-EN	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
MTransE-R	3.0	8.9	2.7	10.2	3.3	14.6
MTransE-PR	32.8	57.6	31.0	56.1	18.9	44.3
BootEA-R	55.2	70.0	47.8	67.7	36.8	58.5
BootEA-PR	45.3	85.4	41.4	79.8	30.2	60.4
GCN-PR	66.2	82.8	60.9	81.5	38.2	52.8
GCN-JR	70.2	82.8	63.9	81.8	42.0	53.8
HGCN-PR	69.3	84.5	63.1	81.3	41.5	54.3
HGCN-JR	70.3	85.4	65.0	83.6	42.5	56.6
RNM	80.6	87.1	74.5	84.6	49.5	62.5

Table 4: Performance on relation alignment

(Wu et al. 2019b), where -R denotes the original model for relation alignment, -PR denotes the model that approximates the relation representations using entity embeddings as in (Wu et al. 2019b), and -JR denotes the model that jointly learns the embeddings of both entities and relations. From the results we can observe that the proposed model RNM performs better than all the baselines especially in Hits@1. Among the baselines, BootEA achieves better results compared with MTransE due to its bootstrapping strategy, while GCN further improves the performance by incorporating the semantic information. The proposed model RNM outperforms the best baseline model (HGCN-JR) by 10.3%, 9.5%, and 7.0% in Hits@1 on the three datasets, respectively. The reason may be that RNM aligns relations by matching the head and the tail entities which can provide more evidence for relation alignment. Moreover, these results confirm our assumption that the entity alignment and the relation alignment can reinforce each other in our model.

Analysis Figure 3 shows the results of entity alignment and relation alignment with different numbers of iterations for the proposed model RNM. With the increase of itera-

tions, the performance of RNM on entity alignment and relation alignment raise accordingly. This proves the effectiveness of the iterative framework of RNM and the assumption that the entity alignment task and the relation alignment task can reinforce each other for better performance.

Since RNM can implement the alignments in a semi-supervised manner, we conduct several experiments on the entity alignment task with different proportions of seed alignments, and results are shown in Figure 4. We choose NMN, which performs the best among baselines, as the comparison model. From the results, we can observe that RNM outperforms NMN in all situations. Even RNM with only 10% seed alignments performs better than the NMN with 40% seed alignments on all the three datasets. This is because RNM explores the useful information of relations when matching neighborhood, and the iterative strategy help to enhance the performance.

Conclusion and Future Work

In this paper, we propose a novel relation-aware neighborhood matching model named RNM for entity alignment. In the model, we jointly learn the embeddings of entities and relations. Moreover, we propose to make use of the semantic information and mapping properties of relations for better entity alignment. In addition, we implement entity alignment and relation alignment iteratively to reinforce each other in a semi-supervised manner. Finally, we evaluate the proposed model on three cross-lingual KG datasets and empirical results demonstrate the effectiveness of RNM.

In the future, we will study how to make use of the side information such as attributes (Zhang et al. 2019) and descriptions (Yang et al. 2019) to improve the accuracy of entity alignment.

Acknowledgments

This work was supported by the National Key Research and Development Project (Grant No.2020AAA0106600) and Peking University Education Big Data Project (Grant No.2020YBC10).

References

- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, 1247–1250.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2787–2795.
- Cao, Y.; Liu, Z.; Li, C.; Li, J.; and Chua, T.-S. 2019a. Multi-Channel Graph Neural Network for Entity Alignment. In *Proceedings of ACL*, 1452–1461.
- Cao, Y.; Wang, X.; He, X.; Hu, Z.; and Chua, T.-S. 2019b. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of WWW*, 151–161.
- Chen, M.; Tian, Y.; Yang, M.; and Zaniolo, C. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*, 1511–1517.
- Hao, Y.; Zhang, Y.; He, S.; Liu, K.; and Zhao, J. 2016. A Joint Embedding Method for Entity Alignment of Knowledge Bases. In *Proceedings of CCKS*, 3–14.
- Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *Proceedings of WSDM*, 105–113.
- Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.
- Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic web* 6(2): 167–195.
- Li, C.; Cao, Y.; Hou, L.; Shi, J.; Li, J.; and Chua, T.-S. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *Proceedings of EMNLP*, 2723–2732.
- Lin, X.; Yang, H.; Wu, J.; Zhou, C.; and Wang, B. 2019. Guiding cross-lingual entity alignment via adversarial knowledge embedding. In *Proceedings of ICDM*, 429–438.
- Pei, S.; Yu, L.; Hoehndorf, R.; and Zhang, X. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proceedings of WWW*, 3130–3136.
- Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of WWW*, 697–706.
- Sun, Z.; Hu, W.; Zhang, Q.; and Qu, Y. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of IJCAI*, 4396–4402.
- Sun, Z.; Wang, C.; Hu, W.; Chen, M.; Dai, J.; Zhang, W.; and Qu, Y. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of AAAI*, volume 34, 222–229.
- Wang, Z.; Lv, Q.; Lan, X.; and Zhang, Y. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*, 349–357.
- Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Yan, R.; and Zhao, D. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of IJCAI*, 5278–5284.
- Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; and Zhao, D. 2019b. Jointly Learning Entity and Relation Representations for Entity Alignment. In *Proceedings of EMNLP*, 240–249.
- Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; and Zhao, D. 2020. Neighborhood Matching Network for Entity Alignment. In *Proceedings of ACL*, 6477–6487.
- Xiong, C.; Power, R.; and Callan, J. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of WWW*, 1271–1279.
- Xu, K.; Wang, L.; Yu, M.; Feng, Y.; Song, Y.; Wang, Z.; and Yu, D. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *Proceedings of ACL*, 3156–3161.
- Yang, H.-W.; Zou, Y.; Shi, P.; Lu, W.; Lin, J.; and Xu, S. 2019. Aligning Cross-Lingual Entities with Multi-Aspect Information. In *Proceedings of EMNLP*, 4422–4432.
- Yang, K.; Liu, S.; Zhao, J.; Wang, Y.; and Xie, B. 2020. COTSAE: CO-Training of Structure and Attribute Embeddings for Entity Alignment. In *Proceedings of AAAI*, 3025–3032.
- Ye, R.; Li, X.; Fang, Y.; Zang, H.; and Wang, M. 2019. A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment. In *Proceedings of IJCAI*, 4135–4141.
- Zhang, Q.; Sun, Z.; Hu, W.; Chen, M.; Guo, L.; and Qu, Y. 2019. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of IJCAI*, 5429–5435.
- Zhu, H.; Xie, R.; Liu, Z.; and Sun, M. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings. In *Proceedings of IJCAI*, 4258–4264.
- Zhu, Q.; Zhou, X.; Wu, J.; Tan, J.; and Guo, L. 2019. Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs. In *Proceedings of IJCAI*, 1943–1949.