Aligning Entities for Heterogeneous Knowledge Graphs with Relational Graph Convolutional Networks

Abstract

Entity alignment is the task of linking entities and their counterparts among multiple knowledge graphs (KGs). Most recent works rely on knowledge graph embedding methods, such as TransE, to represent entities and their context, which often have difficulties in identifying sparse but discriminative evidence, utilizing attribute values and dealing with unbalanced neighboring context. To address these issues, we present RGCN, a novel entity alignment model which leverages relational graph convolutional networks to better characterize entities by incorporating the neighboring relational structure information and taking attribute values into account. We further introduce layer-wise highway network gates to filter out the propagation of noisy information and ensure effective spread of more informative and discriminative neighborhood information. Experiments on a Chinese entity alignment dataset show that our model can consistently outperform competitive baselines in various conditions.

1 Introduction

Acquiring knowledge from large volumes of unstructured textual data plays an important role in not only the development of Semantic Web, but also the understanding of text content. There have been many works transforming online encyclopedia resources, such as Wikipedia[1], into structured knowledge bases (KBs) in a form of ⟨subject entity, predicate/relation, object⟩ triples, such as DBpedia (**?**; **?**), Freebase (**?**), Yago (**?**) etc. Such kind of KBs can be naturally organized into the form of graphs which we call knowledge graphs (KGs). Although most of those KGs originate from Wikipedia, they are usually created independently, thus often use different expressions and surface forms to indicate equivalent entities and relations, let alone those built from different resources, or even different languages. This further makes it more challenging to achieve knowledge sharing, complementary and integration among different KGs.

One of the key techniques to integrate different KGs is **Entity Alignment**, the task of linking the equivalent entities from different KGs if they refer to the same real-world identity, usually with different surface forms. However, entity alignment is not a trivial task, and the alignment system is often complex (**?**; **?**). Traditional approaches, which generally rely on external information such as hyperlinks in web pages and require costly manual feature construction, are often time consuming and labor intensive (**?**).

Models	**entity**	**relation**	**triple**	**use of value**
JE (**?**)	√			
MTransE (**?**)	√	√	√	
JAPE (**?**)	√	√		√ (type)
ITransE (**?**)	√	√		

Table 1: The use of seed alignment information in each method.

Most recently, many efforts have been devoted to the so-called KG embedding-based approaches, following similar ideas to jointly embed the structures of multiple KGs into a unified vector space by leveraging the pre-aligned entities or relations serving as a bridge. These KG embedding-based approaches all rely on translation-based KG embedding models, such as TransE (**?**), to learn entity representations. However, as listed in Table 1, most of these KG embedding-based methods require high-quality seed alignment data, such as pre-aligned KG predicates/relations or triples. Since the predicates/relations or triples in heterogeneous KGs may be very different, it is often difficult and expensive to collect such high-quality alignment data. In addition, as shown by the last column in the Table 1, most of these models, except JAPE (**?**), ignore specific attribute values in the KGs because of their complexity and heterogeneity. And actually, JAPE only considers the types of those values for simplicity, and the specific attribute values, such as *1.86m* as someone's height, are ignored. We argue that those attribute value information are crucial for entity alignment and should be taken into account.

Moreover, although TransE can effectively capture the structure information of KGs, it may not perform well when the neighboring relational structures of two entities are significantly different, e.g., the different sizes of their neighbors, the different relations and neighbor entities. And actually, when we judge whether two entities refer to the equivalent identity, we often only pay attention to the more informative and discriminative neighbors of the two entities, especially when the available clues are sparse.

[1] https://www.wikipedia.org

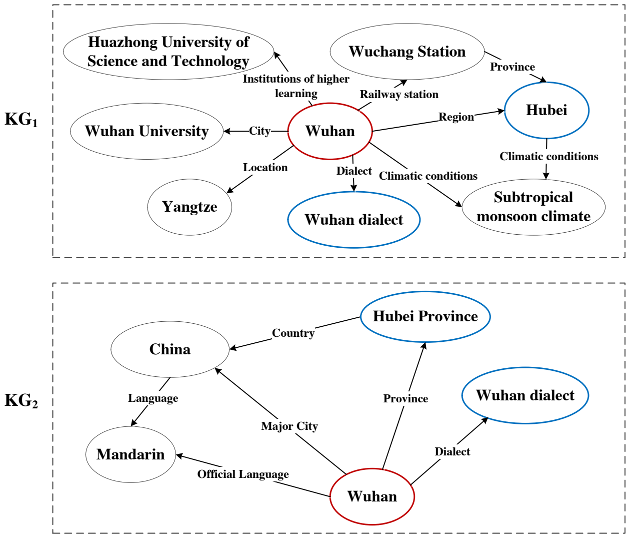

Figure 1: Parts of the neighborhoods of *Wuhan* in two KGs.The red circles indicate the equivalent entities and the blue circles represent the intersection of their neighbors.

In this paper, we propose a new embedding-based entity alignment approach which leverages relational graph convolutional networks to better embed the highly multi-relational structure information of heterogeneous KGs with a set of pre-aligned entities and considers the specific attribute values in multiple KGs. Our model solves the limitations of existing embedding-based methods which ignore the value information and can not properly align those entities of which the neighborhoods are very different or the available clues for alignment are sparse.

More specifically, the contributions of this paper can be summarized as follows:

- We propose RGCN, a novel entity alignment model, to better characterize entities in heterogeneous KGs by considering their neighboring relational structure as well as the attribute values in the KGs.
- We show that layer-wise highway gates play a significant role to control the balance of how much neighborhood information should be passed to a node in our HRGCN model.
- We built a large-scale entity alignment dataset in Chinese, containing 57,240 entities, 3,563 relations, 28,595 attributes and 593,483 triples. We manually aligned 16,969 entity pairs as the gold standards of entity alignment.

2 Problem Formulation

We first introduce the notations used in this paper. A knowledge graph is formalized as $KG = (E, V, R, A, T)$, where E, V, R, A, T are the sets of entities, values, relations, attributes and triples respectively. KG consists of knowledge triples described as (s, p, o), in which $s \in E$, $p \in R$ and $o \in E$ or $p \in A$ and $o \in V$.

The task of entity alignment aims to link entities and their counterparts among multiple KGs. Suppose there are two knowledge graphs $KG_1 = (E_1, V_1, R_1, A_1, T_1)$ and $KG_2 = (E_2, V_2, R_2, A_2, T_2)$, and $\mathbb{L} = \{(e_i^1, e_i^2)|e_i^1 \in E_1, e_i^2 \in E_2, i \in \{1, ..., n\}\}$ is a set of pre-aligned entity pairs between KG_1 and KG_2. The task of entity alignment is to automatically discover new entity alignments based on known alignment seeds.

3 Preliminary

Graph Convolutional Networks

Here we briefly introduce Graph Convolutional Networks (GCNs) (**?**). The inputs of a GCN model are a set of node features and the adjacency matrix of the undirected graph. The output is a new set of node features, containing information of neighbor nodes. GCNs can convolve the kth-order neighborhood information by stacking multiple convolutional layers.

Formally, consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges, respectively. A GCN layer $l + 1$ takes current node features $h^{(l)} = \{h_1^{(l)}, h_2^{(l)}, ..., h_n^{(l)}\}, h_i^{(l)} \in \mathbb{R}^{F^{(l)}}$ and the adjacency matrix A as input, and produces new node representations, $h^{(l+1)} = \{h_1^{(l+1)}, h_2^{(l+1)}, ..., h_n^{(l+1)}\}, h_i^{(l+1)} \in \mathbb{R}^{F^{(l+1)}}$, as output. Here $F^{(l)}$ represents the size of node features in the lth layer. $n = |\mathcal{V}|$, indicates the number of nodes. The layer-wise propagation rule can be simply expressed as:

$$h_i^{(l+1)} = \sigma(\sum_{j \in \mathcal{N}_i} \hat{A}_{ij} h_j^{(l)} W^{(l)} + b^{(l)}) \tag{1}$$

where $\hat{A} = \hat{D}^{-\frac{1}{2}}(A + I)\hat{D}^{-\frac{1}{2}}$, $\hat{D}$ is the degree matrix of $A + I$. $\sigma(\cdot)$ is an activation function. $\mathcal{N}_i$ denotes the set of neighbor indices of node i in the graph, including node i, and $W^{(l)}$ and $b^{(l)}$ are a weight matrix and a bias of layer l.

Relational Graph Convolutional Networks

GCNs have been shown to be very effective at accumulating and encoding features from local, structured neighborhoods. Inspired by these architectures, a simple propagation model for relational (directed and labeled) multi-graphs, R-GCNs (**?**), has been proposed:

$$h_i^{(l+1)} = \sigma(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}) \tag{2}$$

where $\mathcal{N}_i^r$ is the set of neighbor indices of node i under relation $r \in \mathcal{R}$. $c_{i,r}$ is a normalization constant. $W_r^{(l)}$ is the weight matrix corresponding to relation r in layer l and $W_0^{(l)}$ denotes a special relation type, a single self-connection, to each node. (**?**) use a unique one-hot vector for each node in the graph as the input to the first R-GCN layer. And we improve this featureless approach with pre-defined node feature vectors in our RGCN entity alignment model . For R-GCN, multiple layers can be stacked to capture more neighboring characteristics across several relational steps.

To solve the issue about the rapid growth in number of parameters with the number of relations when applying this

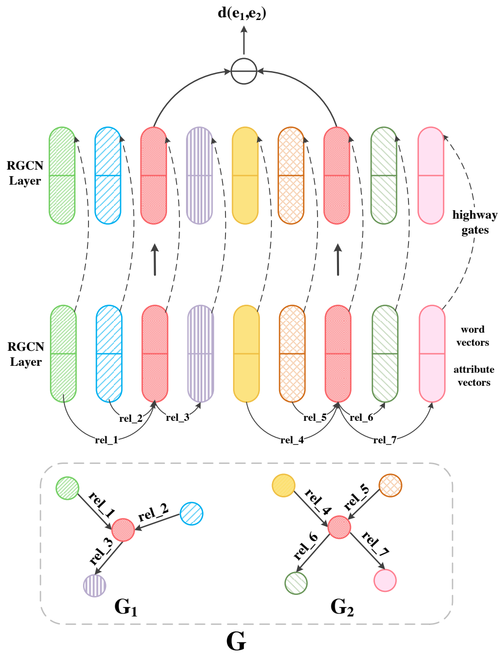

Figure 2: Overall architecture of our HRGCN entity alignment model.

propagation model, (?) provide two methods for regularizing the weights of R-GCN layers: basis- and block-diagonal-decomposition. In our entity alignment model, we choose the basis decomposition which will be introduced in detail in Section 4.2.

4 Approach

In this section, we present detailed description of our entity alignment model. Without loss of generality, we introduce our approach based on two heterogeneous knowledge graphs: $G_1 = (E_1, V_1, R_1, A_1, T_1)$ and $G_2 = (E_2, V_2, R_2, A_2, T_2)$ for entity alignment, where E, V, R, A, T represent entities, values, relations, attributes and triples respectively. We put G_1 and G_2 together in one large graph G. We utilize pre-aligned entity pairs to train our models and then discover new equivalent entities. Figure 2 demonstrates the overall architecture of our model.

4.1 Node Representations

We combine node semantic information with value information to construct input node feature vectors. The specific construction method is as follows.

Semantic Information: We believe that the entities and their counterparts in multiple KGs should be semantically similar. Therefore we leverage the pre-trained word embeddings to introduce the semantic information about the surface forms of entities. We use word2vec software of Tomas Mikolov and his colleagues[2] to generate word embeddings. In our experiments, the window size is 5 and threshold for downsampling the frequent words is 20. Sentences in Baidu baike are used as training data and 155,837 100-dimensional word vectors are generated.

Value Information: For attribute values, we distinguish four kinds of abstract range types, i.e., Integer, Double, Date and String (as default). In this paper, we only consider the first three types, i.e., Integer, Double and Date. We overlook String type values by reason of their complexity and heterogeneity in different KGs.

We construct normalized attribute vector for each entity. Specifically, the dimension of attribute vector is equal to the number of distinct attributes of which the value types belong to Integer, Double and Date in the KG. The elements in an entitys attribute vector equal to the normalized values of the corresponding attributes. If an entity does not have an attribute, the element corresponding to this attribute in the vector is then set to 0.

We concatenate pre-trained word vectors with attribute vectors as the input feature vectors of entities (nodes) in KGs.

4.2 RGCN

The input to our RGCN model are two parts. The first part is the node feature matrix $X^{(0)} \in \mathbb{R}^{N\times d^{(0)}}$ of G, where N is the number of nodes and $d^{(0)}$ is the dimension of the input representations. We utilize predefined node features described in Section 4.1 to construct X instead of using a featureless approach in R-GCNs (?). The second part is the list of adjacency matrixs $A = \{A_1, A_2, ..., A_R | A_i \in \mathbb{R}^{N\times N}\}$, which describes R different relations. We extract R_0 original relations from knowledge graphs, then we add reverse relations in order to pass information from the opposite direction; and add the self loop to retain information of the node itself. These together compose $R = 2R_0 + 1$ relations. In each layer l, the input is $X^{(l-1)} = \{x_1^{(l-1)}, x_2^{(l-1)}, ..., x_N^{(l-1)} | x_i^{(l-1)} \in \mathbb{R}^{d^{(l-1)}}\}$. The forward propagation is formulated as:

$$x_i^{(l+1)} = \mathrm{ReLU}(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{|N_i^r|}W_r^{(l)}x_j^{(l)}) \tag{3}$$

Here $W_r^{(l)} \in \mathbb{R}^{d^{(l)}\times d^{(l-1)}}$ is the weight matrix of relation r. N_i^r is the set of neighbor indices of node i under adjacency matrix $\hat{A}_r$. $\hat{A}_r$ is an approximate of spectral convolutions on A^r, introduced by (?):

$$\hat{A}_r = \hat{D}_r^{-\frac{1}{2}}(A_r + I)\hat{D}_r^{-\frac{1}{2}} \tag{4}$$

where $(\hat{D}_r)_{jj} = \sum_k (A_r + I)_{jk}$.

We get the new embedding matrix $X^{(l+1)} \in \mathbb{R}^{N\times d^{(l+!)}}$ by stacking the output $x_i^{(l+1)}$ together.

As there are generally thousands of relation types in knowledge graphs, there will be a large amount of parameters to train and the model is likely to overfit. Hence we employ the basis decomposition, which is introduced in (?), to regularize the weights:

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)} \tag{5}$$

[2] https://code.google.com/archive/p/word2vec

where $V_b^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ and $a_{rb}^{(l)}$ is the coefficient of matrix $V_b^{(l)}$ for relation r.

Highway RGCN While stacking RGCN layers makes our model capable of learning more neighborhood information from several relational steps, it may as well bring noise from the exponentially increasing neighbors. To reduce the effect of noise and ensure effective spread of more informative and discriminative neighborhood information, we add layer-wise gates similar to highway networks (**?**) to our RGCN entity alignment model. (**?**) have successfully introduced highway gates to GCNs (**?**) to solve the user geolocation problem. We introduce layer-wise highway gates to our RGCN model to finally get Highway RGCN (HRGCN) model and the output of a HRGCN layer is computed as:

$$\begin{aligned} T(x^{(l)}) &= \sigma(W_T^{(l)} x^{(l)} + b_T^{(l)}) \\ x^{(l+1)} &= x^{(l+1)} \cdot T(x^{(l)}) + x^{(l)} \cdot (1 - T(x^{(l)})) \end{aligned} \tag{6}$$

where σ indicates the sigmoid activation function, $\cdot$ is element-wise multiplication, $W_T^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ and $b_T^{(l)} \in \mathbb{R}^{d^{(l)} \times 1}$ are the weight matrix and bias vector of transform gate $T(x^{(l)})$.

4.3 Alignment Prediction

After HRGCN layers, we get the hidden representations $\bar{X}$ of all nodes in both KGs. We measure the similarity between e_1 in G_1 and e_2 in G_2 by the distance between their hidden representations: $d(e_1, e_2) = |\bar{x_{e_1}} - \bar{x_{e_2}}|$, where $|\cdot|$ indicates the l_1 norm. The distance for equivalent entities is expected to be smaller than non-equivalent ones. In our experiments, for a entity e_1 in G_1, we computes the distances between e_1 and all the entities in G_2.

A set of pre-aligned entity pairs $\mathbb{L}$ and the set of negative pairs $\mathbb{L}'$ constructed by corrupting (p, q), i.e. replacing p or q with a randomly chosen entity in G_1 or G_2 are used for training. To maximize the distance between positive and negative instances, we use the margin-based loss function:

$$L = \sum_{(p,q) \in \mathbb{L}} \sum_{(p',q') \in \mathbb{L}'} \max\{0, d(p_i, q_i) - d(p'_i, q'_i) + \gamma\} \tag{7}$$

$\gamma > 0$ is a margin hyper-parameter separating positive and negative entity alignments.

5 Experiments

Our experiments are designed to investigate whether RGCN-based models can better represent an entity with its highly multi-relational structure information compared to KG embedding-based methods, and whether highway gates can effectively capture discriminative neighboring information by controlling the balance of how much neighborhood information should be passed to a node in RGCN.

5.1 Datasets

We evaluate our proposed models and all baselines on two Chinese KGs, Wiki and Baidu.

	Wiki	Baidu	Total
#Entities	25,139	32,101	57,240
#Relations	1,301	2,262	3,563
#Attributes	8,332	20,263	28,595
#Triples	147,139	446,344	593,483

Table 2: The statistics for WBD.

Wiki: We use the Chinese version of the Wikipedia dump[3] released on June 2, 2015 and extract more than 1million entities and 3 million ⟨subject, predicate, object⟩ triples from the Infobox on each page to form the KG.

Baidu: We build a KG based on one of the largest online Chinese encyclopedia, BaiduBaike, by extracting ⟨subject, predicate, object⟩ triples from the Infobox on each page. The resulting KG consists of 11.55 million entities and 39.2 million structured triples.

Based on the two KGs, we manually aligned 16,969 entity pairs as the gold standards of entity alignment, and built a new dataset, called WBD, for entity alignment in Chinese KGs.

5.2 Baselines

JE: As shown in Table 1, JE (**?**) requires the similar seed alignments, pre-aligned entities, as our model, which also performs well among all existing embedding-based models. We use our best effort to implement this model as they do not release any source code or software currently.

ITransE′: ITransE (**?**) is another representative embedding-based methods for entity alignment. However, ITransE requires all relations to be shared between two KGs. Since our WBD dataset was built based on two completely heterogeneous KGs, sharing same relations is unrealistic for our dataset. Thus we build ITransE′, a variant of ITransE which does not use same relations between KGs.

GCNs: We also build a GCN-based model as our baseline which utilize GCNs (**?**) to embed the structure information of two KGs and also takes our predefined node features as input node feature matrix. In our experiments, we stack two layers of GCN.

Variants of our model: Our full model is two-layered Highway RGCN (HRGCN). We also build a no-highway variant, two-layered RGCN. We build another variant HRGCN (w/o X) which does not use the predefined feature vectors mentioned in Section 4.1 as the input to our model.

5.3 Experimental Setup

For all the compared approaches, we use 50% of the gold standards for training and 50% of them for testing. We use Hits@k as the evaluation metrics to assess the performance of all the approaches. Hits@k measures the proportion of correctly aligned entities ranked in the top k. For GCN layers, we set 64 hidden units. To avoid overfitting, we apply L2 regularization with $\lambda = 0.00001$ and utilize dropout with $\mathrm{dropoutrate} = 0.1$. For all variants of our model, we use

[3] https://dumps.wikimedia.org/zhwiki

Models	Hits@1	Hits@10	Hits@50
JE	10.8	21.6	31.2
ITransE′	25.5	34.5	46.9
GCN	23.2	36.3	48.8
RGCN	34.8	49.2	61.5
HRGCN (w/o X)	21.1	30.7	42.7
HRGCN	**65.1**	**71.7**	**76.3**

Table 3: Results comparison of entity alignment.

RGCN with 16 hidden units for each layer and $B = 80$ for basis function decomposition. We apply L2 regularization with $\lambda = 5e-4$ to avoid overfitting. For both GCNs and all variants of our model, we trained with Adam (**?**) for a maximum of 200 epochs using a learning rate of 0.01. All models are initialized using Glorot initialization (**?**).

As mentioned in Section 4.1, we utilize pre-trained 100-dimensional word vectors. By counting the frequency of attributes appearing in each KG in WBD dataset, we select 63 high-frequency attributes from each KG to construct 63-dimensional attribute vectors.

5.4 Results and Discussion

We summarize the performances of all models on the WBD dataset in Table 3.

As shown in Table 3, we can see that ITransE′ outperforms JE regarding all the Hits@k measures and outperforms GCN for Hits@1. This indicates that ITransE is an outstanding model for entity alignment and it also shows that TransE can effectively embed the structure information of KGs which plays an important role in entity alignment. However, GCN-based model performs better than ITransE in Hits@10 and Hits@50. As aforementioned, GCNs leverage convolutional layers to characterize an entity through careful investigations about its neighbors, including both neighboring entities and attribute values, which can provide more fine-grained and accurate modeling and representation for the target entity. Comparing with GCN, the RGCN-based model further boosts the performance by 11.6%, 12.9% and 12.7% for Hits@1, Hits@10 and Hits@50. It shows that introducing highly multi-relational information to GCN framework can achieve significant improvements on KG embedding.

Among all models, when enhanced with layer-wise highway gates, our HRGCN model performs the best, significantly improving upon RGCN by 30.3%, 22.5% and 14.8% for Hits@1, Hits@10 and Hits@50. This indicates that highway gates play a significant role in our model.

When comparing our full model HRGCN with HRGCN (w/o X), we find that removing the predefined input feature matrix X leads to a drop of 44.0% for Hits@1, 41.0% for Hits@10 and 33.6% for Hits@50. This confirms that initializing entity representations using pre-trained word embeddings and normalized value vectors is very helpful in aligning entities from different KGs.

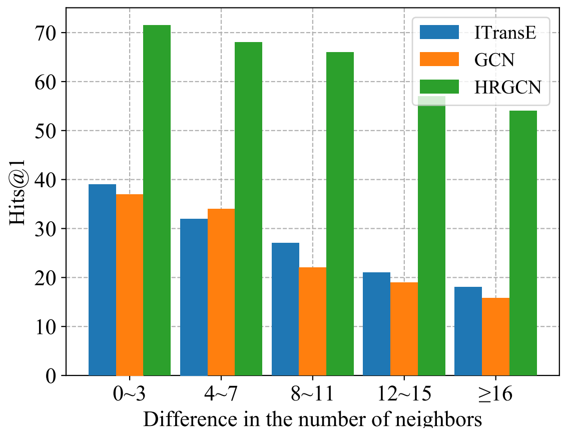

Figure 3: Hits@1 of ITransE, GCN and HRGCN on the five subsets. 0∼3 denotes the subset in which the number of neighbors differs from 0 to 3 for each entity pair, and similar for the remaining subsets.

Analysis We further provide a detailed analysis about the experimental results.

We divide the test set into five subsets according to the difference between the number of neighbors of each entity pair, and compare the performance (in Hits@1) of ITransE, GCN and HRGCN on the five subsets.

As shown in Figure 3, we can see that when the number of neighbors differs by no more than 3, all three models perform well, but when the difference between the entity pairs' neighborhoods becomes more prominent, our HRGCN model tends to deliver more clear improvement.

We randomly choose some examples that our HRGCN model can correctly align but ITransE fails in Table 4, as well as their neighbor information, i.e., number of neighbors, number of values, number of potentially overlapped neighbors or values for each pair. We can find that although several entities have dozens of neighbors in their corresponding KGs, but their similar or overlapped neighbors are quite few, showing again that the available clues for entity alignment are sparse and ITransE may not perform well in this circumstance, while our HRGCN model can still identify useful structure information from those limited clues.

We can also observe from Table 4 that values play an important role in entity alignment. For instance, in the entity pair about *Hubei Province*, nearly half of the neighbors for each entity are values, and among all 5 similar neighbors, 3 of them are are actually values, which provide crucial supporting evidence for the final prediction. Unfortunately, ITransE does not utilize those information, thus is unable to collect sufficient evidence.

In Figure 3, we can see that our HRGCN wins GCN in every subset. This is mainly because introducing highly multi-relational information to GCN with highway gates can help our model better embed the relational structure information and focus on the most discriminative aspects from the tar-

Aligned Entities	#Neighbors Wiki & Baidu	#Similar Neighbors	#Values Wiki & Baidu	#Similar Values
Deng Jiaxian	10 & 33	5	3 & 11	2
Hubei Province	21 & 50	5	10 & 19	3
European Union	66 & 35	6	18 & 8	2
Confucius	10 & 20	4	7 & 3	2

Table 4: The statistics of example entity pairs, which our HRGCN model correctly aligns but ITransE fails.

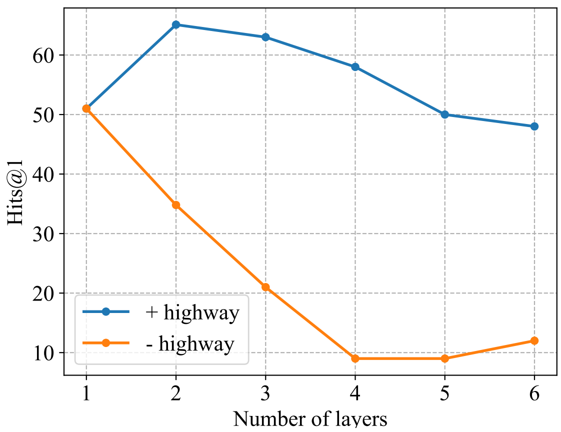

Figure 4: The effect of adding more RGCN layers in terms of Hits@1 over the test set of WBD with and without the highway gates.

get entity's neighbors, thus lead to more accurate representations.

Adding more HRGCN layers can help the center entities obtain information from neighbors that are multiple hops away. However, it might also introduce noisy information from the exponentially increasing neighbors, leading to significant decline in performance as shown in Figure 4 when no highway gates are used. We can observe that the performance of two-layered RGCNs with highway gates improves upon one-layered RGCN. Then by adding more layers the performance of highway RGCNs decreased slowly, but much slower than RGCNs without gates. This confirms that the highway gates effectively control the required balance of neighbor information transmission in RGCNs.

6 Related Work

Our work mainly involves the relational graph convolutional networks (R-GCNs) (**?**). The R-GCN model is an extension of graph convolutional networks (GCNs) (**?**) to large-scale relational data. Then, we will introduce them in detail and provide a systematic review of current work on entity alignment.

Graph Convolutional Networks

Recently, there is an increasing interest in extending neural networks to deal with arbitrarily structured graphs and there have been many encouraging works. Among them, GCNs (**?**; **?**; **?**), a recent class of multilayer neural networks operating on graphs, have been successfully applied to semi-supervised node classification (**?**), semantic role labeling (**?**), neural machine translation (**?**) and so on. For every node in the graph, GCN encodes relevant information about its neighborhood and it is concerned with the adjacent node features.

Relational Graph Convolutional Networks

Since GCNs (**?**) generally operate on undirected and unlabeled graphs, R-GCNs (**?**) are developed specifically for relational (directed and labeled) multi-graphs to deal with the highly multi-relational data characteristic of realistic knowledge graphs. And this model has been successfully exploited in two standard knowledge base completion tasks: Link prediction and entity classification (**?**). In this paper, we successfully construct the entity alignment model which utilizes R-GCNs to embed entities of multiple KGs into a unified vector space.

Entity Alignment

As we mentioned in Section 1, the conventional entity alignment approaches are usually time-consuming and laborious since that the traditional works generally rely on external information and require costly manual feature construction. For example, in the work of Wang Xuepeng et al. (**?**), they need to collect various network semantic labels such as category labels, attribute labels and unstructured text keywords of the entity entries to build a number of semantic similarity calculation models.

To address these issues, several embedding-based methods have been proposed and achieve promising results, such as JE (**?**), MTransE (**?**), JAPE (**?**) and ITransE (**?**).

Following the energy-based framework in TransE, JE jointly learns the embeddings of multiple KGs in a uniform vector space to align entities in KGs by adding the loss of alignment part to the global loss function. In the learning process, JE requires the seed aligned entities share the same embeddings. Although TransE can effectively capture the structure information of KGs, there are still several tough situations where TransE can not perform very well. Since TransE utilizes the relation between the head entity and the tail entity to define the distance between the head entity vector and the tail entity vector, the TransE-based approaches actually tend to require that the neighboring structures of aligned entities should be as similar as possible. Due to the incompleteness of knowledge graphs, the densities of the neighborhoods of the two entities e_1 and e_2 that we need to align may be very different or there are few similar neighbors between two entities, which leads to sparse available clues for alignment (Figure 1 gives an example), which will make a large difference between the learned vectors of e_1 and e_2 by TransE.

Since utilizing TransE to embed entities, MTransE, JAPE and ITransE have the same problems as JE. In addition, besides the pre-aligned entities, MTransE needs a set of triples to be aligned in advance and ITransE needs pre-aligned relations. JAPE needs both relations and attributes to be aligned. Actually, the seed alignments required by MTransE, ITransE and JAPE are difficult to obtain in practice.

These embedding-based approaches all ignored the value information except JAPE. However, values are actually very significant parts of KGs, especially for low-quality KGs in which the entity linking is insufficient. Those low-quality KGs may contain large-scale values and the approaches that do not consider values will lose this part of the information when aligning KGs. JAPE is a joint attribute-preserving embedding model for cross-lingual entity alignment. And JAPE proposed attribute embedding to represent the attribute correlations of KGs which considers the value information, only the type of values.

Instead of utilizing TransE, our approach leverages R-GCNs (**?**) to better characterize entities by incorporating the neighboring relational structure information and considers the value information in multiple KGs.

7 Conclusions

In this paper, we propose a novel neural framework for entity alignment over heterogeneous KGs, which leverages relational graph convolutional networks to better characterize entities when the available clues for alignment are sparse by considering the neighboring relational structures as well as attribute value information. We further utilize highway network gates to enable our model to control the amount of useful neighborhood information expansion. Experiments shows that our solutions can provide more accurate and discriminative entity representation in various situations for high-quality entity alignment.

In the future, we will take KG relations or predicates into consideration to explore richer semantics for both entity alignment and predicate alignment simultaneously. And we also plan to extend our framework into cross-lingual KG alignment, which is more challenging than our current setup.