Aligning Entities for Heterogeneous Knowledge Graphs: A Relational Graph Convolutional Network Based Approach

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 real-world datasets show that our approach further improves entity alignment performance on various KGs, and gets the best performance compared with competitive baselines.

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 has been an extensive body of work in transforming online encyclopedia resources, such as Wikipedia, into structured knowledge bases (KBs) in a form of ⟨*subject entity*, *predicate/relation*, *object*⟩ triples, such as DBpedia (Bizer et al. 2009; Auer et al. 2007), Freebase (Bollacker, Cook, and Tufts 2007), Yago (Suchanek, Kasneci, and Weikum 2008) 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 (Gokhale et al. 2014; Scharffe, Zamazal, and Fensel 2014). 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 (Zhu et al. 2017).

Models	entity	relation	triple	use of value
JE (Hao et al. 2016)	√			
MTransE (Chen et al. 2016)			√	
JAPE (Sun, Hu, and Li 2017)	√	√		√ (type)
ITransE (Zhu et al. 2017)	√	√		

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 with 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 (Bordes et al. 2013), 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 (Sun, Hu, and Li 2017), 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. However, values are actually very significant parts of KGs, especially for low-quality KGs which may contain large-scale values. The approaches that do not consider values will lose this part of information when aligning KGs. 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

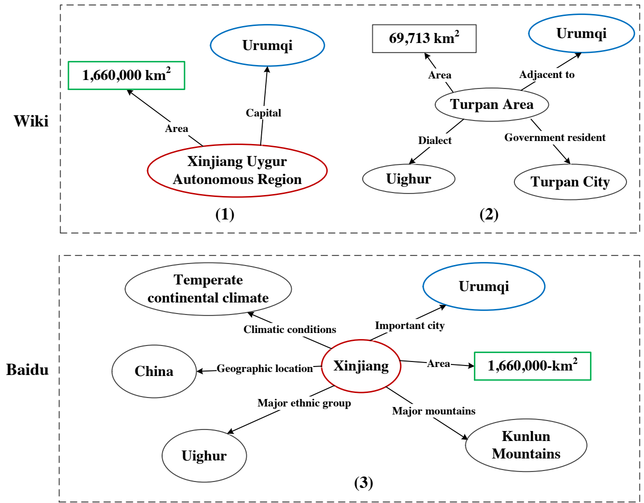

Figure 1: The neighborhoods of three entities extracted from Wiki and Baidu KGs. The red circles indicate the aligned center entities. The blue circles represent the similar neighbors. The green rectangles indicate the same attribute values.

structure information of KGs, it may not perform well when the neighboring relational structures of two entities are significantly different. 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 the equivalent entities from different KGs should be as similar as possible. Nevertheless, 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 similar relations and neighbors between e_1 and e_2 may be few, which leads to sparse available clues for alignment and will make a large difference between the learned vectors of e_1 and e_2 by TransE. But 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.

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 the entities with very different neighborhoods.

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 build a large-scale entity alignment dataset in Chinese, containing 57,240 entities, 3,563 relations, 28,595 attributes, 231,003 relation triples and 515,065 attribute triples. We manually aligned 16,969 entity pairs as the gold standards of entity alignment.

2 Motivation

As a motivation example, consider Figure 1 which depicts the heterogeneous neighborhoods of three center entities extracted from the Wiki and Baidu datasets (see Section 4.1). In the diagram, an entity is marked by a circle and a value is indicated by a rectangle, while an attribute or relation is listed on the edge.

The three neighborhoods have some entities (e.g., *Urumqi*) and attributes (e.g., *Area*) in common but their graph structure is different. Specifically, (1) and (3) contain a common center entity of *Xinjiang* (the surface form of this entity in Wiki is *Xinjiang Uygur Autonomous Region*). Therefore, a successful entity alignment strategy needs to link *Xinjiang* from both KGs. However, state-of-the-art entity alignment methods (Hao et al. 2016; Chen et al. 2016; Sun, Hu, and Li 2017; Zhu et al. 2017) built upon TransE all fail to align this entity because of two reasons: (i) they are misled by some of the redundant entities and attributes that only appear in the Baidu dataset and (ii) they do not utilize attributes like *1,660,000*km^2. This is a problem when using translation-based embedding.

If we look closely into (1) and (3), we find that the *Xinjiang* entity can be aligned using the entity *Urumqi* highlighted by the blue circle together with an attribute value *1,660,000*km^2 highlighted by a green rectangle. In this example, if we can treat other entities and attributes as noise, we can successfully align the *Xinjiang* entity.

This example shows that entity alignment requires one to carefully evaluate the importance of each entity and attribute from heterogeneous KGs. Unfortunately, this information varies across datasets and entities. Because manually obtaining this information for every target entity would incur significant overhead, there is a critical need to automate the process. In this paper, we describe an novel approach to offer this capability based on the R-GCN.

3 Our Approach

Our entity alignment model can be applied to two arbitrary (heterogeneous) KGs. Without loss of generality, we introduce our approach using two KGs: $G_1 = (E_1, V_1, R_1, A_1, T_1)$ 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.

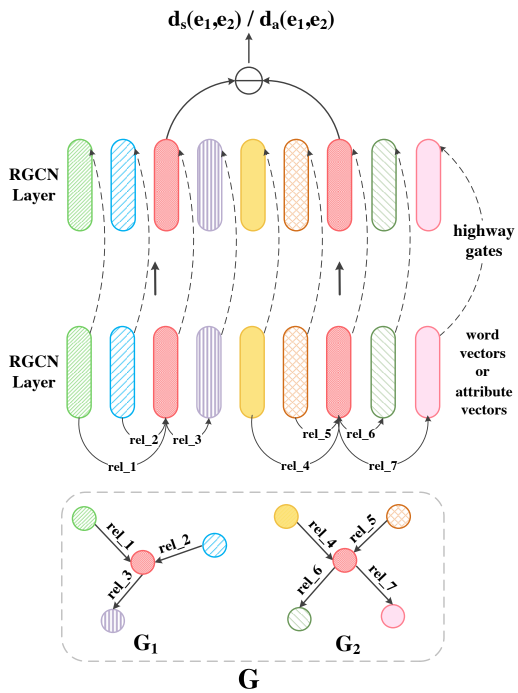

Figure 2: Overall architecture of our entity alignment model.

3.1 Base Model

Our approach is based on the recently proposed relational graph convolutional network (R-GCN) (Schlichtkrull et al. 2017). R-GCN is an extension of Graph Convolutional Networks (GCNs) that operate on local graph neighborhoods (Duvenaud et al. 2015; Kipf and Welling 2016) to large-scale relational data. A R-GCN takes a set of adjacency matrices as input, and produces a new set of node features. Each input adjacency matrix describes the adjacency relationships among all nodes in the graph under each different relation. We choose to use R-GCN because it can model relational (directed and labeled) multi-graphs like KGs. Our work improves the featureless approach of R-GCN with predefined node feature vectors. We believe that in addition to the internal structures, the semantic information of entity names and the attribute information of entities can help R-GCN better embed KGs. Therefore, we incorporate the aforementioned information into the node features as part of the model inputs.

3.2 Node Representations

There are two methods to initialize the node feature vectors. One way is to use the semantics of entity names, and the other is to use entities' attribute information. The specific initialization methods are as follows.

Semantic initialization. We believe that the names of 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 names of entities. For Chinese datasets, we use word2vec software of Tomas Mikolov and his colleagues[1] 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 4,200,006 100-dimensional word vectors are generated.

Attribute initialization. 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.

Since it is the first attempt, we only consider the numerical part of a value, regardless of the unit, that is, we do not insist on normalizing *1.80m* and *180cm*. We leave this for future work.

3.3 RGCN

Here, we present detailed description of our RGCN entity alignment model.

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 3.2 to construct X instead of using a featureless approach in R-GCNs (Schlichtkrull et al. 2017). The second part is the list of adjacency matrices $A = \{A_1, A_2, ..., A_R | A_i \in \mathbb{R}^{N\times N}\}$, which describes the adjacency relationships among N nodes under 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 $X^{(l)} = \{x_1^{(l)}, x_2^{(l)}, ..., x_N^{(l)} | x_i^{(l)} \in \mathbb{R}^{d^{(l)}}\}$. The forward propagation is formulated as:

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

Different from general GCNs, R-GCNs introduce relation-specific transformations which depend on the type and direction of an edge. And this kind of propagation model can better characterize highly multi-relational data characteristic of realistic KGs. Here $W_r^{(l)} \in \mathbb{R}^{d^{(l+1)}\times d^{(l)}}$ is the weight matrix of relation r. N_i^r is the set of neighbor indices of node i under relation r, according to normalized adjacency matrix $\hat{A}_r$. $\hat{A}_r$ is an approximate of spectral convolutions on A^r, introduced by (Kipf and Welling 2016):

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

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

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

We get the new embedding matrix $X^{(l+1)} \in \mathbb{R}^{N \times d^{(l+1)}}$ 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 (Schlichtkrull et al. 2017), to regularize the weights:

$$W_r^{(l)} = \sum_{b=1}^{B} a_{rb}^{(l)} V_b^{(l)} \tag{3}$$

where $V_b^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ 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 (Srivastava, Greff, and Schmidhuber 2015) to our RGCN entity alignment model. (Rahimi, Cohn, and Baldwin 2018) have successfully introduced highway gates to GCNs (Kipf and Welling 2016) 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:

$$\begin{aligned} T(x^{(l)}) &= \sigma(W_T^{(l)} x^{(l)} + b_T^{(l)}) \\ x^{(l+1)} &= x^{(l+1)} \cdot T(x^{(l)}) + x^{(l)} \cdot (1 - T(x^{(l)})) \end{aligned} \tag{4}$$

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

3.4 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:

$$d(e_1, e_2) = |\bar{x}_{e_1} - \bar{x}_{e_2}| \tag{5}$$

, 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. The alignment process can also be reversed, i.e. from G_2 to G_1. Since, we report the results of both directions of entity alignment in our experiments.

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:

$$L = \sum_{(p,q)\in\mathbb{L}} \sum_{(p',q')\in\mathbb{L}'} \max\{0, d(p,q) - d(p',q') + \gamma\} \tag{6}$$

	Entities	Relations	Attributes	Rel. triples	Attr. triples
French	66,858	1,379	4,547	192,191	528,665
English	105,889	2,209	6,422	278,590	576,543

Table 2: The statistics for $DBP15K_{FR-EN}$.

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

3.5 Combination of Semantic Embedding and Attribute Embedding

As we introduced in Section 3.2, there are two methods to initialize the input node feature vectors. When we leverage the pre-trained word embeddings to initialize the node feature vectors, we call HRGCNs acting semantic embedding. And when the nodes are initialized by attribute vectors, we call HRGCNs performing attribute embedding. Following the same architecture in Figure 2, we do semantic embedding and attribute embedding for G respectively.

We integrate semantic embedding and attribute embedding by defining a combined distance D for aligning entities:

$$D(e_1, e_2) = \frac{1}{m}[\omega d_s(e_1, e_2) + (1 - \omega) d_a(e_1, e_2)] \tag{7}$$

where m is the dimension of the new node features produced by HRGCNs. In our experiments, we set the output dimensions of two models (one for semantic embedding and the other one for attribute embedding) to be the same. ω is a hyper-parameter. d_s and d_a are the distance computed by Eq. 5 according to semantic embedding and attribute embedding, respectively.

4 Experimental Setup

Our experiments are designed to investigate whether R-GCN-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 R-GCN.

4.1 Datasets

In our experiments, we use two real-world datasets: DBP15K which was built by (Sun, Hu, and Li 2017) and a Chinese dataset WBD built by ourselves.

DBP15K is a cross-lingual dataset built based on English, Chinese, Japanese and French versions of DBpedia. This dataset has a series of subsets, each of which contains data from two KGs in different languages and provides 15,000 pre-aligned entity pairs. We use the French-English subset of DBP15K in our experiments, because it seems to be the most challenging subset for entity alignment. According to the experimental results of (Sun, Hu, and Li 2017), all previous entity alignment methods perform the worst on this subset. Table 2 shows the statistics of the French-English subset.

	Entities	Relations	Attributes	Rel. triples	Attr. triples
Wiki	25,139	1,301	8,332	87,775	211,949
Baidu	32,101	2,262	20,263	143,228	303,116

Table 3: The statistics for WBD.

For the Chinese dataset, we extracted data from two web-based encyclopedia sources: Wikipedia and Baidu Baike (which is the largest Chinese-language web-based encyclopedia). The two sources contain a rich of information and are widely used to constructed KGs.

Raw Wiki dataset. We used the Chinese version of the Wikipedia dump[2] released on June 2, 2015. From the dump, we extracted over one million entities and three million ⟨subject, predicate, object⟩ triples from the infobox (i.e., a fixed-format table that summarizes some unifying aspect of a Wikipedia article) from each page to form the Wiki KG.

Raw Baidu dataset. From Baidu Baike, we extracted 11.55 million entities and 39.2 million structured triples from the infobox on each page to construct the Baidu KG. The data were collected from Baidu Baike between February 2018 and July 2018.

Evaluation dataset. From the two aforementioned KGs, we manually aligned 16,969 randomly chosen entity pairs as the gold standards of entity alignment. Our strategy to extract evaluation dataset is that we randomly selected an aligned entity pair and then extracted relation and attribute triples for selected entities. We refer this evaluation dataset as WBD which is used to train and evaluate all approaches. The statistics of WBD are listed in Table 3.

4.2 Competitive Approaches

On our WBD dataset, we compare our approach against two state-of-the-art entity alignment methods JE (Hao et al. 2016) and ITransE (Zhu et al. 2017). For JE, we use our best effort to implement this model as they do not release any source code or software currently. As listed in 1, 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.

We also build two GCN-based models as our baselines: GCN and HGCN, which respectively replace the RGCN layers of our RGCN model and HRGCN model with the GCN layers (Kipf and Welling 2016) to embed the structure information of two KGs. And the two GCN-based model combine the semantic embedding and attribute embedding with $\omega = 0.95$ in the combined distance. In our experiments, we stack two GCN layers, where each GCN contains 64 hidden units. To avoid over-fitting, we apply L2 regularization with $\lambda = 0.00001$ and utilize dropout with $\mathrm{dropoutrate} = 0.1$. These hyperparameters were found to give the best overall performance in our experiments.

Additionally, we divide our model into three variants: two-layered Highway RGCN (HRGCN), two-layered no-highway RGCN and HRGCN (w/o X) which does not use the predefined feature vectors mentioned in Section 3.2 as the input to HRGCN. For all variants of our model, we use RGCN with 16 hidden units for each layer and $B = 80$ for basis function decomposition. We set $\omega = 0.95$ in the combined distance and apply L2 regularization with $\lambda = 5e-4$ to avoid overfitting.

On the $DBP15K_{FR-EN}$ dataset, we compare our HRGCN model with JE, MTransE (Chen et al. 2016) and JAPE (Sun, Hu, and Li 2017). Different from WBD dataset, we use RGCN with 32 hidden units for each layer and $B = 100$ for basis function decomposition. The hyper-parameter ω in the combined distance is set to 0.9.

4.3 Evaluation Methodology

For WBD dataset, we break the gold standards for two parts with equal-size, and use one for training and the other for testing. As mentioned in Section 3.2, we utilize pre-trained 100-dimensional Chinese word vectors. By counting the frequency of attributes appearing in each KG of the WBD dataset, we select 63 high-frequency attributes from each KG to construct 63-dimensional attribute vectors.

For $DBP15K_{FR-EN}$ dataset, we use the same split of gold standards for training (30%) and testing (70%) as in (Sun, Hu, and Li 2017). For semantic embedding, we first directly utilize Google Translate [3] to translate French entities and relations into English. And we use pre-trained English word vectors *glove.840B.300d* [4] to construct the input node feature vectors. For attribute embedding, we select 180 high-frequency attributes from each KG and generate 180-dimensional attribute vectors.

We use Hits@k to assess the performance of all the approaches. This metric is widely used for evaluating entity alignment (Hao et al. 2016; Chen et al. 2016; Sun, Hu, and Li 2017; Zhu et al. 2017) by measuring the proportion of correctly aligned entities ranked in the top k. All the neural network based models are trained using the Adam optimizer (Kingma and Ba 2014) for maximum of 200 epochs with a learning rate of 0.01. Furthermore, all the models are initialized using Glorot initialization (Glorot and Bengio 2010).

5 Experiment Results

5.1 Overall Performance

Results on WBD dataset. Table 4 reports the performance of all models on the WBD dataset. Since the performance of each model is very similar in two alignment directions, we will mainly discuss the results of all models in the direction of Baidu→Wiki.

ITransE′ outperforms JE regarding all the Hits@k measures and outperforms GCN for Hits@1 in both alignment directions. 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

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

[3]https://translate.google.cn

[4]http://nlp.stanford.edu/projects/glove/

Models	Baidu → Wiki		Wiki → Baidu	
	Hits@1	Hits@10	Hits@1	Hits@10
JE	10.8	21.6	10.2	20.3
ITransE′	19.4	25.5	17.6	24.8
GCN	17.1	27.3	15.7	25.9
HGCN	19.5	29.6	19.4	29.5
RGCN	37.3	51.5	34.8	51.6
HRGCN (w/o X)	15.0	21.7	14.5	21.5
HRGCN	**58.3**	**67.1**	**57.8**	**65.7**

Table 4: Results comparison of entity alignment on WBD dataset.

$\mathbf{DBP15K_{FR-EN}}$	FR → EN		EN → FR	
	Hits@1	Hits@10	Hits@1	Hits@10
JE	15.38	38.84	14.61	37.25
MTransE	24.41	55.55	21.26	50.60
JAPE (SE)	29.63	64.55	26.55	60.30
JAPE (SE+AE)	32.39	**66.68**	32.97	**65.91**
HRGCN (SE)	33.43	35.72	31.22	34.65
HRGCN (SE+AE)	**44.82**	56.68	**41.08**	52.95

Table 5: Results comparison of entity alignment on $DBP15K_{FR-EN}$ dataset.

which plays an important role in entity alignment. However, GCN-based model performs better than ITransE in Hits@10. 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 20.2% and 24.2% for Hits@1 and Hits@10. 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 21.0% and 15.6% for Hits@1 and Hits@10. This indicates that highway gates play a significant role in our model. And HGCN model which adds highway gates to GCN also improves the performance of GCN by 2.4% and 2.3% for Hits@1 and Hits@10. It suggests that highway gates are also beneficial to GCN.

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 43.3% for Hits@1 and 45.4% for Hits@10. This confirms that initializing entity representations using pre-trained word embeddings and normalized value vectors is very helpful in aligning entities from different KGs.

Results on $\mathbf{DBP15K_{FR-EN}}$ dataset. Table 5 reports Hits@1 and Hits@10 of all the compared approaches on $DBP15K_{FR-EN}$ dataset. Since we use the same French-English subset and the same split of gold standards for training and testing as in (Sun, Hu, and Li 2017), the results of JE, MTransE, and JAPE are obtained from (Sun, Hu, and Li 2017). JAPE has two variants: Structure Embedding (SE), Structure and attribute joint embedding (SE+AE). We also evaluate two variants of our model: HRGCN (SE) which performs semantic embedding with pre-trained word vectors, HRGCN (SE+AE) which combines semantic embedding and attribute embedding. We can see that our full model HRGCN (SE+AE) gets the best Hits@1 in both alignment directions. This indicates that our model has strong noise immunity. Although we handle cross-lingual data through rough machine translation which might introduce lots of noise, our model still outperforms all the compared baselines regarding Hits@1. JAPE (SE+AE) outperforms HRGCN (SE+AE) by 10%∼13% for Hits@10. It shows that JAPE is an excellent model for entity alignmentmight. However JAPE needs additional aligned relations and attributes, while our model does not need such high-quality seed alignment data.

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

5.2 Analysis

We further provide a detailed analysis about the experimental results.

We divide the WBD test set into four 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 four 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. From WBD test set, we randomly choose some examples that our HRGCN model can correctly align but ITransE fails in Table 6, 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 6 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 neigh-

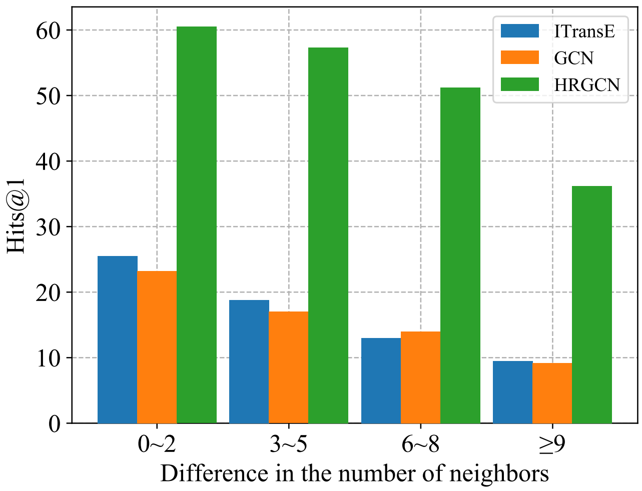

Figure 3: Hits@1 of ITransE, GCN and HRGCN on the four WBD subsets. 0∼2 denotes the subset in which the number of neighbors differs from 0 to 2 for each entity pair, and similar for the remaining subsets.

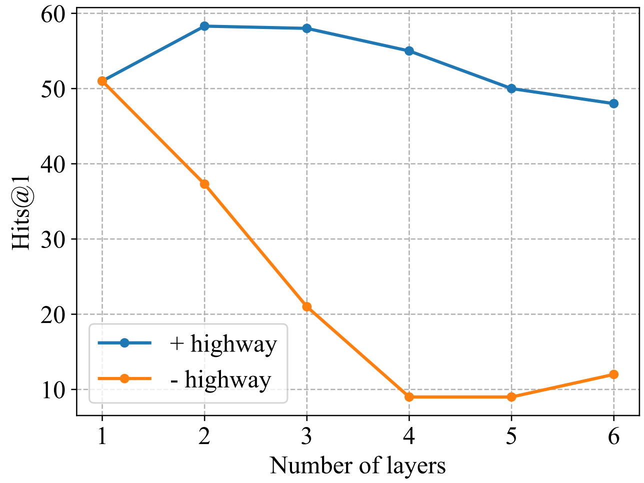

Figure 4: The effect of adding more RGCN layers in terms of Hits@1 over the test set of WBD with and without the highway gates.

bors, 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 target 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 builds upon the following foundations, while qualitatively differing from each.

Graph convolutional networks. Recently, GCNs (Duvenaud et al. 2015; Kearnes et al. 2016) have demonstrated promising results in domains that have previously been dominated by kernel-based methods or graph-based regularization. They are shown to be effective in performing NLP tasks like semi-supervised node classification (Kipf and Welling 2016), semantic role labeling (Marcheggiani and Titov 2017), neural machine translation (Bastings et al. 2017). Our work builds on these past foundations by extending GCNs to process relational data.

Relational graph convolutional networks. Since GCNs generally operate on undirected and unlabeled graphs, R-GCNs (Schlichtkrull et al. 2017) are proposed for relational (directed and labeled) multi-graphs. It has been successfully exploited in two standard knowledge base completion tasks: link prediction and entity classification (Schlichtkrull et al. 2017). However, this technique has not be used for entity alignment and our work is the first to do so.

Entity alignment. As an important NLP task, entity alignment is certainly not a new research topic. Previous approaches of entity alignment typically follow a labour-intensive and time-consuming process to tune model features. For example, the work presented in (Wang et al. 2017) requires one to collect network semantic labels like category labels, attribute labels and unstructured text keywords of the entity entries to build the alignment model.

To address these issues, several embedding-based methods have been proposed and achieve promising results. JE (Hao et al. 2016) jointly learns the embeddings of different KGs in a uniform vector space for entity alignment based on aligned entities. MTransE (Chen et al. 2016) also uses structural information of KGs for cross-lingual KG alignment with known aligned triples. Also for cross-lingual KG alignment, JAPE (Sun, Hu, and Li 2017) proposed a joint attribute-preserving embedding model based on aligned entities, relations and attributes. Moreover, ITransE (Zhu et al. 2017) presented iterative entity alignment via joint knowledge embeddings, requiring all relations being shared among KGs.

Since utilizing TransE to embed entities, JE, MTransE, JAPE and ITransE might not perform well under several tough situations and these embedding-based approaches all ignored the specific attribute values as mentioned in

Section 1. In addition, the seed alignments required by MTransE, JAPE and ITransE are difficult to obtain in practice.

Instead of utilizing TransE, our approach leverages R-GCNs (Schlichtkrull et al. 2017) to better characterize entities by incorporating the neighboring relational structure information and considers the specific 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.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, International Semantic Web Conference, Asian Semantic Web Conference, ISWC 2007 + Aswc 2007, Busan, Korea, November*, 722–735.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; and Sima'An, K. 2017. Graph convolutional encoders for syntax-aware neural machine translation. 1957–1967.

Bizer, C.; Lehmann, J.; Kobilarov, G.; Becker, C.; Hellmann, S.; and Hellmann, S. 2009. Dbpedia - a crystallization point for the web of data. *Web Semantics Science Services & Agents on the World Wide Web* 7(3):154–165.

Bollacker, K.; Cook, R.; and Tufts, P. 2007. Freebase: A shared database of structured general human knowledge. In *AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, 1962–1963.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Chen, M.; Tian, Y.; Yang, M.; and Zaniolo, C. 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*.

Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Hirzel, T.; and Adams, R. P. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *International Conference on Neural Information Processing Systems*, 2224–2232.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research* 9:249–256.

Gokhale, C.; Das, S.; Doan, A.; Naughton, J. F.; Rampalli, N.; Shavlik, J.; and Zhu, X. 2014. Corleone: hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 601–612. ACM.

Hao, Y.; Zhang, Y.; He, S.; Liu, K.; and Zhao, J. 2016. A joint embedding method for entity alignment of knowledge bases. In *China Conference on Knowledge Graph and Semantic Computing*, 3–14. Springer.

Kearnes, S.; Mccloskey, K.; Berndl, M.; Pande, V.; and Riley, P. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* 30(8):1–14.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks.

Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. 1506–1515.

Rahimi, A.; Cohn, T.; and Baldwin, T. 2018. Semi-supervised user geolocation via graph convolutional networks.

Scharffe, F.; Zamazal, O.; and Fensel, D. 2014. Ontology alignment design patterns. *Knowledge and information systems* 40(1):1–28.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. V. D.; Titov, I.; and Welling, M. 2017. Modeling relational data with graph convolutional networks.

Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *Computer Science*.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics Science Services & Agents on the World Wide Web* 6(3):203–217.

Sun, Z.; Hu, W.; and Li, C. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, 628–644. Springer.

Wang, X.; Liu, K.; He, S.; Liu, S.; Zhang, Y.; and Zhao, J. 2017. Multi-source knowledge bases entity alignment by leveraging semantic tags. *Chinese Journal of Computers* 40(3):701–711.

Zhu, H.; Xie, R.; Liu, Z.; and Sun, M. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4258–4264. AAAI Press.