Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs

Abstract

Entity alignment is the task of linking entities with the same real-world identity from different knowledge graphs (KGs), which has been recently dominated by embedding-based methods. Such approaches work by learning KG representations so that entity alignment can be performed by measuring the similarities between two entity embeddings. While promising, prior work in the field often fails to properly capture complex relation information that commonly exists in multi-relational KGs, leaving much room for improvement. In this paper, we propose a novel relation-aware dual-graph convolutional network (RDGCN) to incorporate relation information via attentive interactions between the knowledge graph and its dual relation counterpart, and further capture neighboring structures to learn better entity representations. Experiments on three real-world cross-lingual datasets show that our approach delivers better and more robust results over the state-of-the-art alignment methods by learning better KG representations.

1 Introduction

Knowledge graphs (KGs) transform unstructured knowledge into simple and clear triples of $\langle$*head, relation, tail*$\rangle$ for rapid response and reasoning of knowledge. KGs are the building blocks for many applications like information retrieval [Dalton *et al.*, 2014], recommendation systems [Catherine and Cohen, 2016], question-answering [Cui *et al.*, 2017], etc. To support various applications, there is considerable work on knowledge representation learning to construct distributed representations for both entities and relations. Exemplary works for embedding-based approaches are the so called *trans-family* methods like TransE [Bordes *et al.*, 2013], TransH [Wang *et al.*, 2014], TransR [Lin *et al.*, 2015b] and PTransE [Lin *et al.*, 2015a], which interpret a relation as the translation operating on the embeddings of its head entity and tail entity.

However, KGs are usually incomplete, and different KGs are often complementary to each other. This makes a compelling case to design a technique that can integrate heterogeneous knowledge among different KGs. An effective way for doing this is **Entity Alignment**. There have been existing efforts devoted to embed different KGs towards entity alignment. Most of them, like JE [Hao *et al.*, 2016], MTransE [Chen *et al.*, 2017], JAPE [Sun *et al.*, 2017], IPTransE [Zhu *et al.*, 2017] and BootEA [Sun *et al.*, 2018], rely on *trans-family* models to learn entity representations according to a set of prior alignments. The most recent work [Wang *et al.*, 2018], takes a different approach by utilizing the Graph Convolutional Networks (GCNs) [Kipf and Welling, 2017] to jointly represent multiple KG entities, showing a new, promising direction for entity alignment.

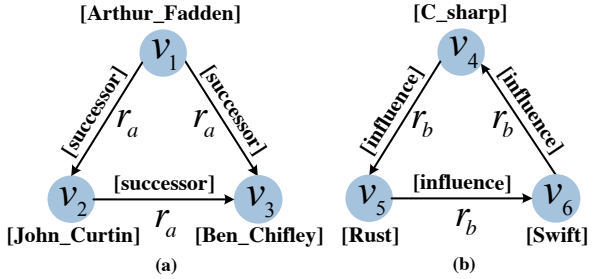

Figure 1: Examples of triangular structures.

Compared with conventional feature based methods [Wang *et al.*, 2017], embedding-based methods have the advantage of requiring less human involvement in feature construction, allowing entity alignment methods to be scaled to large KGs. However, there are still several hurdles that prevent a wider adoption of embedding-based approaches. First, as mentioned above, most existing methods use *trans-family* models as the backbone to embed KGs, which are constrained by the assumption $head + relation \approx tail$. This strong assumption makes it inefficient for the model to capture more complex relation information in multi-relational graphs.

As a motivation example, Figure 1 shows a real-world example from the *DBP15K*$_{ZH-EN}$ [Sun *et al.*, 2017] (one of the real-world KGs used in our experiments). Prior study [Li *et al.*, 2018b] shows that trans-family methods cannot capture the triangular structures depicted in the diagram. For instance, for the structure of Figure 1(a), TransE requires the forms of $v_1 + r_a \approx v_2$, $v_2 + r_a \approx v_3$ and $v_1 + r_a \approx v_3$ to hold at the same time. However, to satisfy the former two equations, we would have $v_1 + 2r_a \approx v_3$, which is contradictory to the third equation $v_1 + r_a \approx v_3$. Accordingly, the alignment performance will inevitably be compromised if the KG

representations are learned with the trans-family, since more complex structures such as triangular ones frequently appear in multi-relational graphs.

GCN-based model [Wang *et al.*, 2018] represents a leap forward for embedding-based entity alignment. However, this approach is also unable to properly model relation information. Since the vanilla GCN operates on the undirected and unlabeled graphs, GCN-based model would ignore the useful relation information of KGs. Although the relational graph convolutional networks (RGCNs) [Schlichtkrull *et al.*, 2017] could be used to model multi-relational graphs, a RGCN simply employs one weight matrix for each relation and would require an excessive set of parameters for real-world KGs that often contain thousands of relations, making it extremely difficult to learn an effective model. Dual-Primal Graph Convolutional Networks (DPGCNN) [Monti *et al.*, 2018] offer a new solution for the problem. DPGCNNs alternate convolution operations on the graph and its dual graph, whose vertices correspond to the edges of the original graph, and iteratively apply an graph attention mechanism to enhance primal edge representations using its dual graph. Compared with GCN and RGCN, DPGCNN can better explore complex edge structures and produce better KG presentations.

Inspired by DPGCNN, in this paper, we propose a novel relation-aware dual-graph convolutional network (RDGCN) to tackle the challenge of proper capturing and integration for relation information. While DPGCNN serves a good starting point, applying it to learn KG representations is not trivial. Doing so requires us to find a way to better approximate relation representations and characterize the relationship between different KG relations. We address this by extending the DPGCNN to develop a weighted model, and explore the head/tail representations initialized with entity names as a proxy to capture relation information without excessive model parameters that are often hard to train.

As a departure from GCNs and RGCNS, our RDGCN allows multiple rounds of interactions between the primal entity graph and its dual relation graph, enabling the model to effectively incorporate more complex relation information into entity representations. To further integrate neighboring structural information, we also extend GCNs with highway gates.

We evaluate our RDGCN on three real-world datasets. Experimental results show that RDGCN can effectively address the challenges mentioned above and significantly outperforms 6 recently proposed approaches on all datasets. The key contribution of this work is a novel DPGCNN-based model for learning robust KG representations. Our work is the first to extend DPGCNNs for entity alignment, which yields significantly better performance over the state-of-the-art alternatives.

2 Related Work

Graph Convolutional Networks. Recently, there has been an increasing interest in extending neural networks to deal with graphs. There have been many encouraging works which are often categorized as spectral approaches [Bruna *et al.*, 2013; Henaff *et al.*, 2015; Defferrard *et al.*, 2016; Kipf and Welling, 2017] and spatial approaches [Atwood and Towsley, 2016; Hamilton *et al.*, 2017; Veličković *et al.*, 2018]. The Graph Convolutional Networks (GCNs) [Kipf and Welling, 2017] have recently emerged as a powerful deep learning-based approach for many natural language processing tasks like semi-supervised node classification [Kipf and Welling, 2017], semantic role labeling [Marcheggiani and Titov, 2017] and neural machine translation [Bastings *et al.*, 2017]. Furthermore, as an extension of GCNs, the relational graph convolutional networks (RGCNs) [Schlichtkrull *et al.*, 2017] have recently been proposed to model relational data and have been successfully exploited in link prediction and entity classification. Recently, the graph attention networks (GATs) [Veličković *et al.*, 2018] have been proposed and achieve the state-of-the-art performance. The DPGCNN [Monti *et al.*, 2018] discussed in Section 1 generalizes GAT model and achieves better performance on vertex classification, link prediction, and graph-guided matrix completion tasks.

Inspired by the capability of DPGCNN on determining neighborhood-aware edge features, we propose the first relation-aware multi-graph learning framework for entity alignment.

Entity Alignment. 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. Recently, embedding-based methods [Hao *et al.*, 2016; Chen *et al.*, 2017; Sun *et al.*, 2017; Zhu *et al.*, 2017; Sun *et al.*, 2018; Wang *et al.*, 2018] have been proposed to address this issue. In addition, NTAM is a non-translational approach that utilizes a probabilistic model for the alignment task [Li *et al.*, 2018a]. KDCoE is a semi-supervised learning approach for co-training multilingual KG embeddings and the embeddings of entity descriptions [Chen *et al.*, 2018].

As a departure from prior work, our approach directly models the relation information by constructing the dual relation graph. As we will show later in the paper, doing so improves the learned entity embeddings which in turn lead to more accurate alignment.

3 Problem Formulation

Formally, a KG is represented as $G = (E, R, T)$, where E, R, T are the sets of entities, relations and triples, respectively. Let $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$ be two heterogeneou KGs to be aligned. That is, an entity in G_1 may have its counterpart in G_2 in a different language or in different surface names. As a starting point, we can collect a small number of equivalent entity pairs between G_1 and G_2 as the *alignment seeds* $\mathbb{L} = \{(e_{i_1}, e_{i_2}) | e_{i_1} \in E_1, e_{i_2} \in E_2\}$. We define the entity alignment task as automatically finding more equivalent entities using the alignment seeds. Those known aligned entity pairs can be used as training data.

4 Our Approach: RDGCN

In order to better incorporate relation information to the entity representations, given the input KG (i.e., the primal graph),

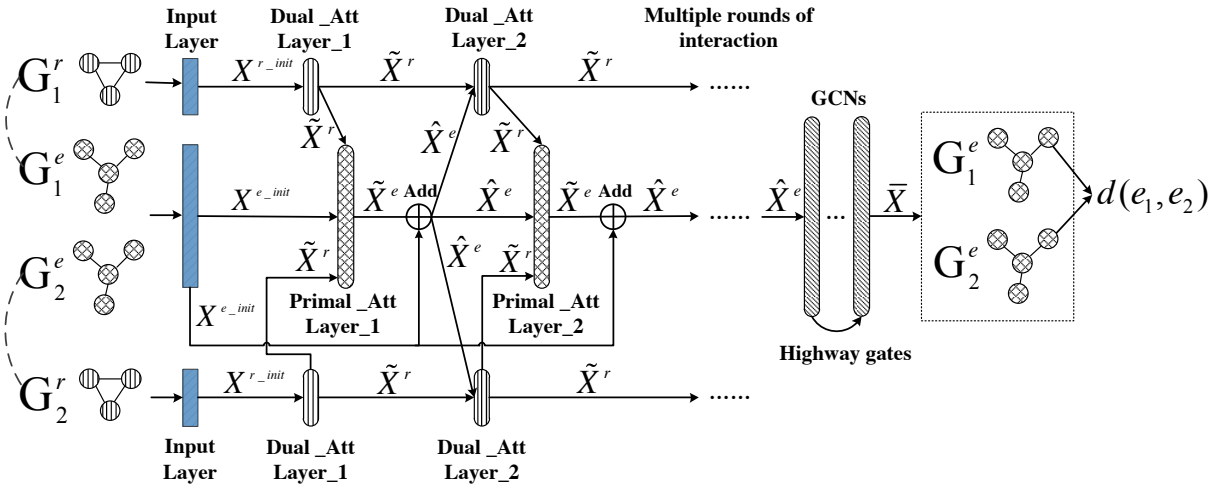

Figure 2: Overall architecture of our RDGCN. G_1^r and G_2^r are the dual relation graphs of G_1^e and G_2^e, respectively. In our RDGCN model, $\mathcal{G}^e$ consists of G_1^e and G_2^e, and $\mathcal{G}^r$ consists of G_1^r and G_2^r.

we first construct its dual relation graph whose vertices denote the relations in the original primal graph, and then, we utilize a graph attention mechanism to encourage interactions between the dual relation graph and the primal graph. The resulting vertex representations in the dual graph are then fed to GCN [Kipf and Welling, 2017] layers with highway gates to capture the neighboring structure information. The final entity representations will be used to determine whether two entities should be aligned. Figure 2 provides an overview architecture of our model.

4.1 Constructing the Dual Relation Graph

Without loss of generality, we put G_1 and G_2 together as the *primal graph* $\mathcal{G}^e = (\mathcal{V}^e, \mathcal{E}^e)$, where the vertex set $\mathcal{V}^e = E_1 \cup E_2$ is the union of all vertices in G_1 and G_2, and the edge set $\mathcal{E}^e = T_1 \cup T_2$ is the union of all undirected edges/triples in G_1 and G_2. Note that we do not connect the alignment seeds in $\mathcal{G}^e$, thus G_1 and G_2 are disconnected in $\mathcal{G}^e$.

Given the primal graph $\mathcal{G}^e$, its *dual relation graph* $\mathcal{G}^r = (\mathcal{V}^r, \mathcal{E}^r)$ is constructed as follows: 1) for each type of relation r in $\mathcal{G}^e$, there will be a vertex v^r in $\mathcal{V}^r$, thus $\mathcal{V}^r = R_1 \cup R_2$; 2) if two relations, r_i and r_j, share the same head entities or tail entities in $\mathcal{G}^e$, then we create an edge u_{ij}^r in $\mathcal{G}^r$ connecting v_i^r and v_j^r.

Different from the original design of dual graph, here we expect the dual relation graph can be more expressive about the relationship between different v^rs in $\mathcal{G}^e$. We thus weight each edge u_{ij}^r in $\mathcal{G}^r$ with a weight w_{ij}^r according to how likely the two relations v_i^r and v_j^r share similar heads or tails in $\mathcal{G}^e$, computed as:

$$w_{ij}^r = H(r_i, r_j) + T(r_i, r_j) \tag{1}$$

$$H(r_i, r_j) = \frac{H_i \cap H_j}{H_i \cup H_j},\ T(r_i, r_j) = \frac{T_i \cap T_j}{T_i \cup T_j} \tag{2}$$

where H_i is the set of head entities for relation r_i in $\mathcal{G}^e$ and T_i is the set of tail entities for r_i in $\mathcal{G}^e$.

4.2 Interactions between Dual and Primal Graphs

Our goal of introducing dual relation graph is to better incorporate relation information into the primal graph representations. To this end, we propose to apply a graph attention mechanism (GAT) to obtain vertex representations for the dual relation graph and the primal graph iteratively, where the attention mechanism helps to prompt interactions between the two graphs. Each dual-primal interaction contains two layers, the dual attention layer and the primal attention layer. Note that we can stack multiple interactions for mutual improvement on both graphs.

Dual Attention Layer. Let $\mathbf{X}^r \in \mathbb{R}^{m \times 2d}$ denote the input dual vertex representation matrix, where each row corresponds to a vertex in the dual relation graph $\mathcal{G}^r$. Different from the vanilla GAT [Veličković *et al.*, 2018], we compute the dual attention scores using the primal vertex features $\hat{\mathbf{X}}^e$ (computed by Eq. 8) produced by the primal attention layer from the previous interaction module:

$$\tilde{\mathbf{x}}_i^r = \sigma^r\Big(\sum_{j \in N_i^r} \alpha_{ij}^r \mathbf{W}^r \mathbf{x}_j^r\Big), \tag{3}$$

$$\alpha_{ij}^r = \frac{exp(w_{ij}^r a^r [\mathbf{W}^r \mathbf{c}_i \| \mathbf{W}^r \mathbf{c}_j])}{\sum_{k \in N_i^r} exp(w_{ik}^r a^r [\mathbf{W}^r \mathbf{c}_i \| \mathbf{W}^r \mathbf{c}_k])}, \tag{4}$$

where $\tilde{\mathbf{x}}_i^r$ denotes the d'-dimensional output representation at dual vertex v_i^r (corresponding to relation $r_i \in \mathcal{G}^e$); $\mathbf{x}_j^r$ denotes the dual representation of vertex v_j^r; N_i^r is the set of neighbor indices of v_i^r; α_{ij}^r is the dual attention score; $\mathbf{W}^r \in \mathbb{R}^{d' \times 2d}$ is a linear transformation applied to every vertex; a^r is a fully connected layer mapping the $2d'$-dimensional input into a scalar; σ^r is the activation function, ReLU; $\|$ is the concatenation operation; $\mathbf{c}_i$ is the relation representation for relation r_i in $\mathcal{G}^e$ obtained from the previous primal attention layer.

Note that within our graph embedding based framework, we are not able to provide relation representations directly, due to limited training data. We thus approximate the relation representation for r_i by concatenating its averaged head and

tail entity representations in $\mathcal{G}^e$ as:

$$\mathbf{c}_i = [\frac{\sum_{k \in H_i} \hat{\mathbf{x}}_k^e}{|H_i|} \| \frac{\sum_{l \in T_i} \hat{\mathbf{x}}_l^e}{|T_i|}], \quad (5)$$

where $\hat{\mathbf{x}}_k^e$ and $\hat{\mathbf{x}}_l^e$ are the output representations of the k-th head entity and l-th tail entity of relation r_i from the previous primal attention layer.

A special case is when the current dual attention layer is the first layer of our model, we do not have $\mathbf{x}_j^r$ in Eq. 3 produced by the previous dual attention layer, therefore, use an initial dual vertex representation produced by Eq. 5 with the initial primal vertex representations $\mathbf{X}^{e\_init}$. Similarly, $\mathbf{c}_i$ will be obtained with the initial primal $\mathbf{X}^{e\_init}$ as well.

Primal Attention Layer. In this layer, when applying GAT on the primal graph, we can compute the primal attention scores using the dual vertex representations in $\mathcal{G}^r$, which actually correspond to the relations in the primal graph $\mathcal{G}^e$. In this way, we are able to influence the primal vertex embeddings using the relation representations produced by the dual attention layer.

Specifically, we use $\mathbf{X}^e \in \mathbb{R}^{n \times d}$ to denote the input primal vertex representation matrix. For an entity e_q in primal graph $\mathcal{G}^e$, its representation $\tilde{\mathbf{x}}_q^e$ can be computed by:

$$\tilde{\mathbf{x}}_q^e = \sigma^e(\sum_{t \in N_q^e} \alpha_{qt}^e \mathbf{W}^e \mathbf{x}_t^e), \quad (6)$$

$$\alpha_{qt}^e = \frac{exp(a^e(\tilde{\mathbf{x}}_{qt}^r))}{\sum_{k \in N_q^e} exp(a^e(\tilde{\mathbf{x}}_{qk}^r))}, \quad (7)$$

where $\tilde{\mathbf{x}}_{qt}^r$ denotes the dual representation for r_{qt} (the relation between entity e_q and e_t) obtained from $\mathcal{G}^r$; α_{qt}^e is the primal attention score; N_q^e is the set of neighbor indices of entity e_q in $\mathcal{G}^e$; $\mathbf{W}^e \in \mathbb{R}^{\tilde{d} \times d}$ is a matrix of learnable weights; a^e is a fully connected layer mapping the d'-dimensional input into a scalar and σ^e is the primal layer activation function, ReLU.

In our model, the initial representation matrix for the primal vertices, $\mathbf{X}^{e\_init}$, can be initialized using entity names, which provide important evidence for entity alignment. We therefore preserve the evidence explicitly by mixing the initial representations with the output of primal attention layer:

$$\hat{\mathbf{x}}_q^e = \beta_s * \tilde{\mathbf{x}}_q^e + \mathbf{x}_q^{e\_init}, \quad (8)$$

where $\hat{\mathbf{x}}_q^e$ denotes the final output representation of the interaction module for entity e_q in $\mathcal{G}^e$; β_s is a weighting parameter for the s-th primal attention layer.

4.3 Incorporating Structural Information

After multiple rounds of interaction between the dual relation graph and the primal graph, we are able to collect relation-aware entity representations from the primal graph. Next, we apply two-layer GCNs [Kipf and Welling, 2017] with highway gates to the resulting primal graph to further incorporating evidence from their neighboring structures.

In each GCN layer l with entity representations $X^{(l)}$ as input, the output representations $X^{(l+1)}$ can be computed as:

$$X^{(l+1)} = \mathrm{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^{(l)} W^{(l)}), \quad (9)$$

where $\tilde{A} = A + I$ is the adjacency matrix of the primal graph $\mathcal{G}^e$ with added self-connections and I is an identity matrix; $\tilde{D}_{jj} = \sum_k \tilde{A}_{jk}$ and $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is a layer-specific trainable weight matrix.

In addition, to control the noise accumulated across layers, we introduce layer-wise gates between GCN layers, which is similar in spirit to the highway networks [Srivastava *et al.*, 2015]:

$$T(X^{(l)}) = \sigma(X^{(l)} W_T^{(l)} + b_T^{(l)}), \quad (10)$$

$$X^{(l+1)} = T(X^{(l)}) \cdot X^{(l+1)} + (1 - T(X^{(l)})) \cdot X^{(l)}, \quad (11)$$

where $X^{(l)}$ is the input to layer $l+1$; σ is a sigmoid function; $\cdot$ is element-wise multiplication; $W_T^{(l)}$ and $b_T^{(l)}$ are the weight matrix and bias vector for the transform gate $T(X^{(l)})$, respectively.

Alignment. With the final entity representations $\bar{X}$ collected from the output of GCN layers, entity alignment can be performed by simply measuring the similarity or distance between two entities. Specifically, the distance, $d(e_1, e_2)$, between two entities, e_1 from G_1 and e_2 from G_2 can be calculated as:

$$d(e_1, e_2) = \|\bar{x}_{e_1} - \bar{x}_{e_2}\|_{L_1}. \quad (12)$$

One can definitely use more complex decision functions.

4.4 Training

For training, we expect the distance between aligned entity pairs to be as close as possible, and the distance between negative entity pairs to be as far as possible. We thus utilize a margin-based scoring function as the training objective:

$$L = \sum_{(p,q) \in \mathbb{L}} \sum_{(p',q') \in \mathbb{L}'} \max\{0, d(p,q) - d(p',q') + \gamma\}, \quad (13)$$

where $\gamma > 0$ is a margin hyper-parameter; $\mathbb{L}$ is our alignment seeds and $\mathbb{L}'$ is the set of negative instances.

Rather than random sampling, we look for challenging negative samples to train our model. Given a positive aligned pair (p, q), we choose the $\mathcal{K}$-nearest entities of p (or q) according to Eq. 12 in the embedding space to replace q (or p) as the negative instances.

5 Experimental Setup

Datasets. We evaluate our approach on three large-scale cross-lingual datasets from DBP15K [Sun *et al.*, 2017]. These datasets are built upon Chinese, English, Japanese and French versions of DBpedia. Each dataset contains data from two KGs in different languages and provides 15K pre-aligned entity pairs. Table 1 gives the statistics of the datasets. We use the same training/testing split with previous works [Sun *et al.*, 2018], 30% for training and 70% for testing.

Comparison Models. We compare our approach against 6 more recent embedding-based alignment methods that we have mentioned in Section 1: JE [Hao *et al.*, 2016], MTransE [Chen *et al.*, 2017], JAPE [Sun *et al.*, 2017], IPTransE [Zhu *et al.*, 2017], BootEA [Sun *et al.*, 2018] and GCN [Wang *et al.*, 2018], where the BootEA achieves the best performance on DBP15K.

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

Table 1: Summary of the DBP15K datasets.

Model Variants. To evaluate different components of our model, we provide four implementation variants of RDGCN for ablation studies, including (1) GCN-s: a two-layered GCN with entity name initialization but no highway gates; (2) RGCN-s: a two-layered RGCN [Schlichtkrull *et al.*, 2017] with entity name initialization; (3) HGCN-s: a two-layered GCN with entity name initialization and highway gates; (4) RD: an implementation of two dual-primal interaction modules, but without the subsequent GCN layers.

Implementation Details. The configuration we used is: $\beta_1 = 0.1$, $\beta_2 = 0.3$, and $\gamma = 1.0$. The dimensions of hidden representations in dual and primal attention layers are $d = 300$, $d' = 600$, and $\tilde{d} = 300$. All dimensions of hidden representations in GCN layers are 300. The learning rate is set to 0.001 and we sample $\mathcal{K} = 125$ negative pairs every 10 epochs. In order to utilize entity names in different KGs for better intilialization, we use Google Translate to translate Chinese, Japanese, and French entity names into English, and then use pre-trained English word vectors *glove.840B.300d* [1] to construct the input entity representations for the primal graph. It is worth noting that Google Translate can not guarantee accurate translations for named entities without any context. We manually check 100 English translations for Japanese/Chinese entity names, and find around 20% of English translations as incorrect, posing further challenges for our model.

Metrics. We use Hits@k, a widely used metric [Sun *et al.*, 2018; Wang *et al.*, 2018] in our experiments. A Hits@k score (higher is better) is computed by measuring the proportion of correctly aligned entities ranked in the top k list.

6 Results and Discussion

6.1 Main Results

Table 2 shows the performance of all compared approaches on the evaluation datasets. By using a bootstrapping process to iteratively explore many unlabeled data, BootEA gives the best Hits@10 score on $DBP15K_{ZH-EN}$ and clearly outperforms GCN and other translation-based models, such as JE, MTransE, JAPE, and IPTransE. It is not surprising that GCN outperforms most translation-based models, i.e., JE, MTransE, JAPE and IPTransE. By performing graph convolution over an entity's neighbors, GCN is able to capture more structural characteristics of knowledge graphs, especially when using more GCN layers, while the translation assumption in translation-based models focuses more on the relationship among heads, tails and relations.

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

Models	ZH-EN		JA-EN		FR-EN	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
JE	21.27	42.77	18.92	39.97	15.38	38.84
MTransE	30.83	61.41	27.86	57.45	24.41	55.55
JAPE	41.18	74.46	36.25	68.50	32.39	66.68
IPTransE	40.59	73.47	36.69	69.26	33.30	68.54
BootEA	62.94	**84.75**	62.23	85.39	65.30	87.44
GCN	41.25	74.38	39.91	74.46	37.29	74.49
GCN-s	50.82	79.15	53.09	82.96	54.49	84.73
RGCN-s	46.57	74.29	48.68	77.82	51.11	80.07
HGCN-s	69.65	82.53	75.54	87.87	88.09	95.27
RD	61.81	73.83	68.54	80.22	84.64	91.98
RDGCN	**70.75**	84.55	**76.74**	**89.54**	**88.64**	**95.72**

Table 2: The overall alignment performance for all models on the DBP15K datasets. Numbers in bold indicate the best performance.

We observe that RDGCN gives the best performance across all metrics and datasets, except for Hits@10 on $DBP15K_{ZH-EN}$ where the performance of RDGCN is second to BootEA with a marginally lower score (84.55 vs 84.75). While BootEA serves a strong baseline by showing what can be achieved by exploiting many unlabeled data, our RDGCN has the advantage of requiring less prior alignment data to learn better representations. We believe that a bootstrapping process can further improve the performance of RDGCN, and we leave this for future work. Later in Section 6.3, we show that RDGCN maintains consistent performance and significantly outperforms BootEA when the training dataset size is reduced. The good performance of RDGCN is largely attributed to its capability for learning relation-aware embeddings. Keep in mind that our RDGCN is initialized using machine translated entity names, but still achieves the best performance in almost all settings, demonstrating the effectiveness and robustness of the proposed method for entity alignment.

6.2 Ablation studies

GCN-s vs. GCN: As shown in Table 2, GCN-s considerably improves GCN in all datasets, resulting in a 17.2% increase on Hits@1 on $DBP15K_{FR-EN}$. As mentioned in Section 5, the three cross-lingual datasets require us to handle cross-lingual data through rough machine translations, which is likely to introduce lots of noise ($\sim$80% accuracy in our pilot study). But our improvement over GCN shows that although noisy in nature, those rough translations can still provide useful evidence to capture, thus should not be ignored.

GCN-s vs. RGCN-s: RGCN is an extension of GCN by explicitly modeling the KG relations, but in our experiments, we observe that GCN-s achieves better performance than RGCN-s on all datasets. As discussed in Section 1, RGCN usually requires much more training data to learn an effective model due to its large number of parameters, and the available training data in our evaluation might be not sufficient for fully unlocking the potential of RGCN.

HGCN-s vs. GCN-s: When comparing HGCN-s and GCN-s, we can see that HGCN-s greatly boosts the performance of GCN-s after employing the layer-wise highway gates, e.g.,

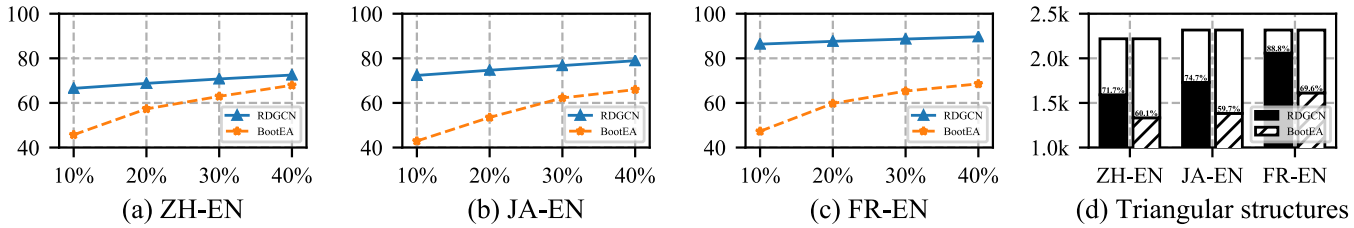

Figure 3: (a), (b) and (c) show the performance of RDGCN and BootEA using different proportions of prior entity alignments on the DBP15K datasets. The x-axes are the proportions of prior alignments, and the y-axies are Hits@1 scores. (d) shows the performance of RDGCN and BootEA on triangular structures. The x-axis is the datasets and y-axis is the number of correctly predicted pairs.

over 30% improvement of Hits@1 on $DBP15K_{FR-EN}$. This is mainly due to their capability of preventing noisy vertices from driving the KG representations.

HGCN-s vs. RDGCN : When comparing HGCN-s with our RDGCN, we can see that the interaction modules performing on the dual relational graph and primal graph are crucial to the performance: removing the dual and primal attention layers leads to a drop of 1.1% on Hits@1 and 2.02% on Hits@10 on $DBP15K_{ZH-EN}$. The interaction modules in our framework can explore the relation characteristics of KGs by introducing the approximate relation information and fully integrate the relation and entity information after multiple interactions between the dual relation and the primal graphs. The results show that effective modeling and use of relation information is beneficial for entity alignment.

RD vs. RDGCN : Comparing RD with RDGCN, there is a significant drop in performance when removing the GCN layers from our model, e.g., the Hits@1 of RD and RDGCN differ by 8.94% on $DBP15K_{ZH-EN}$. This is not surprising, because the dual-primal graph interactions are designed to integrate KG relation information, while the GCN layers can effectively capture the neighboring structural information of KGs. These two key components are, to some extent, complementary to each other, and should be combined together to learn better relation-aware representations.

6.3 Analysis

Triangular structures: Figure 3(d) shows the performance of RDGCN and BootEA, the state-of-the-art alignment model, on the testing instances with triangular structures. We can see that the alignment accuracy of our RDGCN for entities with triangular structures is significantly higher than that of BootEA in all three datasets, showing that RDGCN can better deal with the complex relation information.

Impact of available prior alignments: We further compare our RDGCN with BootEA by varying the proportion of pre-aligned entities from 10% to 40% with a step of 10%. As expected, the results of both models on all three datasets gradually improve with an increased amount of prior alignment information. According to Figure 3(a-c), our RDGCN consistently outperforms BootEA, and seems to be insensitive to the proportion of prior alignments. When only using 10% of the pre-aligned entity pairs as training data, RDGCN still achieves promising results. For example, RDGCN using 10% of prior alignments achieves 86.35% for Hits@1 on $DBP15K_{FR-EN}$. This result translates to a 17.79% higher Hits@1 score over BootEA when BootEA uses 40% of prior alignments. These results further confirm the robustness of our model, especially with limited prior alignments.

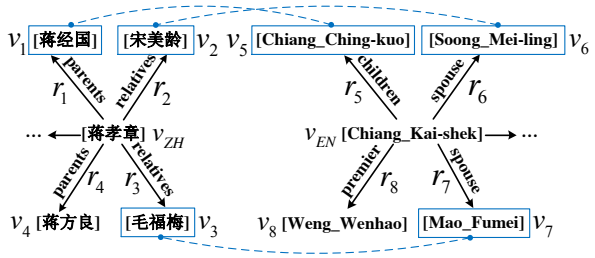

Figure 4: An example in $DBP15K_{ZH-EN}$, where the blue dash lines indicate the connected entities are aligned.

Case Study: Figure 4 shows an example in $DBP15K_{ZH-EN}$ and the target entity pair, (v_{ZH} and v_{EN}), that should not be aligned. The competitive translation-based models, including BootEA, gives lower distance scores for (v_{ZH} and v_{EN}), suggesting that these two entities should be aligned. This is because those models fail to address the specific relation information associated with the three aligned neighboring entities. For this example, both v_1 and v_5 indicate the person *Chiang_Ching-kuo*, but v_1 has the relation *parents* with v_{ZH}, while v_2 has the relation *children* with v_{EN}. Utilizing such information, a better alignment model should produce a larger distance score for the two entities despite they have similar neighbors. By carefully considering the relation information during the dual-primal interactions, our RDGCN gives a larger distance score, leading to the correct alignment result.

7 Conclusions

This paper presents a novel relation-aware dual-graph convolutional network for entity alignment over heterogeneous KGs. Our approach is designed to explore complex relation information that commonly exists in multi-relational KGs. By modeling the attentive interactions between the primal graph and dual relation graph, our model is able to incorporate relation information with neighboring structural information through gated GCN layers, and learn better entity representations for alignment. Compared to the state-of-the-art methods, our model uses less training data but achieves the best alignment performance across three real-world datasets.

References

[Atwood and Towsley, 2016] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *NIPS*, pages 1993–2001, 2016.

[Bastings *et al.*, 2017] Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*, 2017.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Catherine and Cohen, 2016] Rose Catherine and William Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 325–332. ACM, 2016.

[Chen *et al.*, 2017] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, 2017.

[Chen *et al.*, 2018] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. *arXiv preprint arXiv:1806.06478*, 2018.

[Cui *et al.*, 2017] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: learning question answering over qa corpora and knowledge bases. In *VLDB*, pages 565–576, 2017.

[Dalton *et al.*, 2014] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *SIGIR*, pages 365–374. ACM, 2014.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.

[Hamilton *et al.*, 2017] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[Hao *et al.*, 2016] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jian Zhao. A joint embedding method for entity alignment of knowledge bases. In *CCKS*, 2016.

[Henaff *et al.*, 2015] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Li *et al.*, 2018a] Shengnan Li, Xin Li, Rui Ye, Mingzhong Wang, Haiping Su, and Yingzi Ou. Non-translational alignment for multi-relational networks. In *IJCAI*, 2018.

[Li *et al.*, 2018b] Xin Li, Huiting Hong, Lin Liu, and William K Cheung. A structural representation learning for multi-relational networks. *arXiv preprint arXiv:1805.06197*, 2018.

[Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

[Lin *et al.*, 2015b] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187, 2015.

[Marcheggiani and Titov, 2017] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.

[Monti *et al.*, 2018] Federico Monti, Oleksandr Shchur, Aleksandar Bojchevski, Or Litany, Stephan Günnemann, and Michael M Bronstein. Dual-primal graph convolutional networks. *arXiv preprint arXiv:1806.00770*, 2018.

[Schlichtkrull *et al.*, 2017] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*, 2017.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[Sun *et al.*, 2017] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, pages 628–644, 2017.

[Sun *et al.*, 2018] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402, 2018.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[Wang *et al.*, 2017] Xuepeng Wang, Kang Liu, Shizhu He, Shulin Liu, Yuanzhe Zhang, and Jun Zhao. Multi-source knowledge bases entity alignment by leveraging semantic tags. *Chinese Journal of Computers*, 40(3):701–711, 2017.

[Wang *et al.*, 2018] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357, 2018.

[Zhu *et al.*, 2017] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, pages 4258–4264, 2017.