Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs

Yuting Wu¹, Xiao Liu¹, Yansong Feng¹*, Zheng Wang², Rui Yan¹ and Dongyan Zhao¹

¹Institute of Computer Science and Technology, Peking University, China
²School of Computing and Communications, Lancaster University, U. K.

{wyting, lxlisa, fengyansong, ruiyan, zhaodongyan}@pku.edu.cn, z.wang@lancaster.ac.uk

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 entity embeddings. While promising, prior works in the field often fail 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) are the building blocks for many natural language processing applications like question-answering [Zhang et al., 2018], text classification [Wang et al., 2016], recommendation systems [Zhang et al., 2016], etc. Knowledge in KGs is usually organized into triples of \langle head entity, relation, tail entity \rangle. To support various applications, there are considerable works 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], and PTransE [Lin et al., 2015], 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.

Compared with conventional feature based methods [Saraw, et al., 2012; Mahdisoltani et al., 2013], 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 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 DBP15KZH-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 three formulas \( v_1 + r_u \approx v_2, v_2 + r_a \approx v_3 \) 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.

The 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, a GCN-based model would ignore the useful relation information of KGs. Although the relational graph convolutional networks (RGCNs) [Schlichtkrull et al., 2018] could be used to model multi-relational graphs, an 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. This drawback makes it extremely difficult to learn an effective RGCN 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 a graph attention mechanism to enhance primal edge representations using its dual graph. Compared with GCNs and RGCNs, the DPGCNN can better explore complex edge structures and produce better KG representations.

Inspired by the 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 the 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 DPGCNN 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., 2014; Defferrard et al., 2016; Kipf and Welling, 2017] and spatial approaches [Atwood and Towsley, 2016; Hamilton et al., 2017; Velickovic 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., 2018] 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) [Velickovic et al., 2018] have been proposed and achieved 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 usually require intensive expert participation [Sarasua et al., 2012] to design model features [Mahdisoltani et al., 2013] or an external source contributed by other users [Wang et al., 2017]. 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 [Li et al., 2018] [Veličković et al., 2018] is a non-translational approach that utilizes a probabilistic model for the alignment task. KDCoE [Chen et al., 2018] is a semi-supervised learning approach for co-training multilingual KG embeddings and the embeddings of entity descriptions.

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 heterogeneous 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 $L = \{(e_{i1}, e_{i2}) | e_{i1} \in E_1, e_{i2} \in E_2\}$. 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),
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 primal graph are then connected layer mapping the dual vertex representation matrix, where each row corresponds to a vertex in 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.

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

$$\tilde{x}^r_i = \sigma^r(\sum_{j \in N^r_i} \alpha^r_{ij} x^r_j),$$  \hspace{1cm} (3)

$$\alpha^r_{ij} = \frac{\exp(\eta(w^r_{ij} a^r[c_i || c_j]))}{\sum_{k \in N^r_i} \exp(\eta(w^r_{ik} a^r[c_i || c_k]))},$$  \hspace{1cm} (4)

where $\tilde{x}^r_i$ denotes the $d'$-dimensional output representation at dual vertex $v^r_i$ (corresponding to relation $r_i \in G^r$); $x^r_j$ denotes the dual representation of vertex $v^r_j$; $N^r_i$ is the set of neighbor indices of $v^r_i$; $\alpha^r_{ij}$ is the dual attention score; $a^r$ is a fully connected layer mapping the $2d'$-dimensional input into a scalar; $\sigma^r$ is the activation function, ReLU; $\eta$ is the Leaky ReLU; $\parallel$ is the concatenation operation; $c_i$ is the relation representation for relation $r_i$ in $G^r$ 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 $G^e$ as:
\[
c_i = \left[ \sum_{k \in H_i} \hat{x}_k^e \right] \left[ \sum_{l \in T_i} \hat{x}_l^e \right],
\]
(5)
where $\hat{x}_k^e$ and $\hat{x}_l^e$ 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 $x_k^e$ 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 $X^{e,\text{init}}$. Similarly, $c_i$ will be obtained with the initial primal $X^{e,\text{init}}$ 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 $G^e$, which actually correspond to the relations in the primal graph $G^r$. 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 $X^e \in \mathbb{R}^{n \times d}$ to denote the input primal vertex representation matrix. For an entity $e_i$ in primal graph $G^r$, its representation $\hat{x}_q^e$ can be computed by:
\[
\hat{x}_q = \sigma \left( \sum_{t \in N_q^e} \alpha^e_{qt} x_t^e \right),
\]
(6)
\[
\alpha^e_{qt} = \frac{\exp(qt(\hat{x}_q^e))}{\sum_{k \in N_q^e} \exp(qk(\hat{x}_k^e))},
\]
(7)
where $\hat{x}_q^e$ denotes the dual representation for $r_q^e$ (the relation between entity $e_q$ and $e_t$) obtained from $G^r$; $\alpha^e_{qt}$ is the primal attention score; $N_q^e$ is the set of neighbor indices of entity $e_q$ in $G^r$; $a^e$ is a fully connected layer mapping the $d'$-dimensional input into a scalar; $\sigma$ is the primal layer activation function, ReLU and $\eta$ is the Leaky ReLU.

In our model, the initial representation matrix for the primal vertices, $X^{e,\text{init}}$, 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:
\[
\hat{x}_q^e = \beta_s \hat{x}_q + x_q^{e,\text{init}},
\]
(8)
where $\hat{x}_q^e$ denotes the final output representation of the interaction module for entity $e_q$ in $G^e$; $\beta_s$ 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:
\[
X^{(l+1)} = \xi(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X^{(l)} W^{(l)}),
\]
(9)
where $\tilde{A} = A + I$ is the adjacency matrix of the primal graph $G^r$ 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; $\xi$ is the activation function ReLU. We treat $G^e$ as an undirected graph when constructing $\tilde{A}$, in order to allow the information to flow in both directions.

In addition, to control the noise accumulated across layers and preserve useful relation information learned from interactions, we introduce layer-wise gates between GCN layers, which is similar in spirit to the highway networks [Srivastava et al., 2015]:
\[
T(X^{(l)}) = \sigma(X^{(l)} W_T^{(l)} + b_T^{(l)}),
\]
(10)
\[
X^{(l+1)} = T(X^{(l)}) \cdot X^{(l)} + (1 - T(X^{(l)})) \cdot X^{(l)},
\]
(11)
where $X^{(l)}$ is the input to layer $l + 1$; $\sigma$ 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 $\hat{X}$ collected from the output of GCN layers, entity alignment can be performed by simply measuring the 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:
\[
d(e_1, e_2) = \|\hat{x}_{e_1} - \hat{x}_{e_2}\|_{L_1},
\]
(12)

### 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:
\[
L = \sum_{(p,q) \in L} \max\{0, d(p, q) - d(p', q') + \gamma\},
\]
(13)
where $\gamma > 0$ is a margin hyper-parameter; $L$ is our alignment seeds and $L'$ 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 $K$-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], IPTTransE [Zhu et al., 2017], BootEA [Sun et al., 2018] and GCN [Wang et al., 2018], where the BootEA achieves the best performance on DBP15K.
Table 1: Summary of the DBP15K datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Entities</th>
<th>Relations</th>
<th>Rel. triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBP15K_{ZH-EN}</td>
<td>Chinese</td>
<td>66,469</td>
<td>2,830</td>
</tr>
<tr>
<td></td>
<td>English</td>
<td>98,125</td>
<td>2,317</td>
</tr>
<tr>
<td>DBP15K_{JA-EN}</td>
<td>Japanese</td>
<td>65,744</td>
<td>2,043</td>
</tr>
<tr>
<td></td>
<td>English</td>
<td>95,680</td>
<td>2,096</td>
</tr>
<tr>
<td>DBP15K_{FR-EN}</td>
<td>French</td>
<td>66,858</td>
<td>1,379</td>
</tr>
<tr>
<td></td>
<td>English</td>
<td>105,889</td>
<td>2,209</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>ZH-EN</th>
<th>JA-EN</th>
<th>FR-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hits@1</td>
<td>Hits@10</td>
<td>Hits@1</td>
</tr>
<tr>
<td>JE</td>
<td>21.27</td>
<td>42.77</td>
<td>18.92</td>
</tr>
<tr>
<td>MTransE</td>
<td>30.83</td>
<td>61.41</td>
<td>27.86</td>
</tr>
<tr>
<td>JAPE</td>
<td>41.18</td>
<td>74.46</td>
<td>36.25</td>
</tr>
<tr>
<td>IPTransE</td>
<td>40.59</td>
<td>73.47</td>
<td>36.69</td>
</tr>
<tr>
<td>BootEA</td>
<td>62.94</td>
<td>84.75</td>
<td>62.23</td>
</tr>
<tr>
<td>GCN</td>
<td>41.25</td>
<td>74.38</td>
<td>39.91</td>
</tr>
<tr>
<td>GCN-s</td>
<td>50.82</td>
<td>79.15</td>
<td>53.09</td>
</tr>
<tr>
<td>RGCN-s</td>
<td>46.57</td>
<td>74.29</td>
<td>48.68</td>
</tr>
<tr>
<td>HGCN-s</td>
<td>69.65</td>
<td>82.53</td>
<td>75.54</td>
</tr>
<tr>
<td>RD</td>
<td>61.81</td>
<td>73.83</td>
<td>68.54</td>
</tr>
<tr>
<td>RDGCN</td>
<td>70.75</td>
<td>84.55</td>
<td>76.74</td>
</tr>
</tbody>
</table>

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., 2018] 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 \( \hat{d} = 300. \) All dimensions of hidden representations in GCN layers are 300. The learning rate is set to 0.001 and we sample \( K = 125 \) negative pairs every 10 epochs. In order to utilize entity names in different KGs for better initialization, 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. 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/
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 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.

**Case Study** Figure 4 shows an example in DBP15K \(_{ZH-EN}\) and the target entity pair, \((v_{ZH} \text{ and } v_{EN})\), should not be aligned. The competitive translation-based models, including BootEA, give lower distance scores for \((v_{ZH} \text{ and } v_{EN})\), 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 \(v_5\) 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.

### Acknowledgements

This work is supported in part by the NSFC (Grant No. 61672057, 61672058, 61872294), the National Hi-Tech R&D Program of China (No. 2018YFC0831905), and a UK Royal Society International Collaboration Grant (IE161012). For any correspondence, please contact Yansong Feng.
References


[Wang et al., 2016] Chenguang Wang, Yangqiu Song, Hao-ran Li, Ming Zhang, and Jiawei Han. Text classification with heterogeneous information network kernels. In AAAI, 2016.


