

# Jointly Learning Entity and Relation Representations for Entity Alignment

Yuting Wu<sup>1</sup>, Xiao Liu<sup>1</sup>, Yansong Feng<sup>1,2\*</sup>, Zheng Wang<sup>3</sup> and Dongyan Zhao<sup>1,2</sup>

<sup>1</sup>Wangxuan Institute of Computer Technology, Peking University, China

<sup>2</sup>The MOE Key Laboratory of Computational Linguistics, Peking University, China

<sup>3</sup>School of Computing, University of Leeds, U.K.

{wyting, lxlisa, fengyansong, zhaodongyan}@pku.edu.cn  
z.wang5@leeds.ac.uk

## Abstract

Entity alignment is a viable means for integrating heterogeneous knowledge among different knowledge graphs (KGs). Recent developments in the field often take an embedding-based approach to model the structural information of KGs so that entity alignment can be easily performed in the embedding space. However, most existing works do not explicitly utilize useful relation representations to assist in entity alignment, which, as we will show in the paper, is a simple yet effective way for improving entity alignment. This paper presents a novel joint learning framework for entity alignment. At the core of our approach is a Graph Convolutional Network (GCN) based framework for learning both entity and relation representations. Rather than relying on pre-aligned relation seeds to learn relation representations, we first approximate them using entity embeddings learned by the GCN. We then incorporate the relation approximation into entities to iteratively learn better representations for both. Experiments performed on three real-world cross-lingual datasets show that our approach substantially outperforms state-of-the-art entity alignment methods.

## 1 Introduction

Knowledge graphs (KGs) transform unstructured knowledge into simple and clear triples of  $\langle head\ entity, relation, tail\ entity \rangle$  for rapid response and reasoning of knowledge. They are an effective way for supporting various NLP-enabled tasks like machine reading (Yang and Mitchell, 2017), information extraction (Wang et al., 2018a), and question-answering (Zhang et al., 2018b).

Even though many KGs originate from the same resource, e.g., Wikipedia, they are usually created independently. Therefore, different KGs often use

different expressions and surface forms to indicate equivalent entities and relations – let alone those built from different resources or languages. This common problem of heterogeneity makes it difficult to integrate knowledge among different KGs. A powerful technique to address this issue is *Entity Alignment*, the task of linking entities with the same real-world identity from different KGs.

Classical methods for entity alignment typically involve a labor-intensive and time-consuming process of feature construction (Mahdisoltani et al., 2013) or rely on external information constructed by others (Suchanek et al., 2011). Recently, efforts have been devoted to the so-called embedding-based approaches. Representative works of this direction include 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). More recent work (Wang et al., 2018b) uses the Graph Convolutional Network (GCN) (Kipf and Welling, 2017) to jointly embed multiple KGs.

Most of the recent works (e.g., JE, MTransE, JAPE, IPTransE and BootEA) rely on the translation-based models, such as TransE (Bordes et al., 2013), which enable these approaches to encode both entities and relations of KGs. These methods often put more emphasis on the entity embeddings, but do not explicitly utilize relation embeddings to help with entity alignment. Another drawback of such approaches is that they usually rely on pre-aligned relations (JAPE and IPTransE) or triples (MTransE). This limits the scale at which the model can be effectively performed due to the overhead for constructing seed alignments for large KGs. Alternative methods like GCN-based models, unfortunately, cannot directly obtain relation representations, leaving much room for improvement.

Recent studies have shown that jointly modeling entities and relations in a single framework can

\*Corresponding author.

improve tasks like information extraction (Miwa and Bansal, 2016; Bekoulis et al., 2018). We hypothesize that this will be the case for entity alignment too; that is, the rich relation information could be useful for improving entity alignment as entities and their relations are usually closely related. Our experiments show that this is even a conservative target: by jointly learning entity and relation representations, we can promote the results of both entity and relation alignment.

In this work, we aim to build a learning framework that jointly learns entity and relation representations for entity alignment; and we want to achieve this with only a small set of pre-aligned entities but not relations. Doing so will allow us to utilize relation information to improve entity alignment without paying extra cost for constructing seed relation alignments.

Our work is enabled by the recent breakthrough effectiveness of GCNs (Kipf and Welling, 2017) in extracting useful representations from graph structures. Although GCNs provide a good starting point, applying it to develop a practical and efficient framework to accurately capture relation information across KGs is not trivial. Because a vanilla GCN operates on the undirected and unlabeled graphs, a GCN-based model like (Wang et al., 2018b) would ignore the useful relation information of KGs. While the Relational Graph Convolutional Network (R-GCN) (Schlichtkrull et al., 2018) can model multi-relational graphs, existing R-GCNs use a weight matrix for each relation. This means that an R-GCN would require an excessive set of parameters to model thousands of relations in a typical real-world KG, making it difficult to learn an effective model on large KGs.

A key challenge of our joint learning framework is how to generate useful relation representations at the absence of seed relation alignments, and to ensure the framework can scale to a large number of types of relations. We achieve this by first approximating the relation representations using entity embeddings learned through a small amount of seed entity alignments. We go further by constructing a new joint entity representation consisting of both relation information and neighboring structural information of an entity. The joint representations allow us to iteratively improve the model’s capability of generating better entity and relation representations, which lead to not only better entity alignment, but also more accurate re-

lation alignment as a by-product.

We evaluate our approach by applying it to three real-world datasets. Experimental results show that our approach delivers better and more robust results when compared with state-of-the-art methods for entity and relation alignments. The key contribution of this paper is a novel joint learning model for entity and relation alignments. Our approach reduces the human involvement and the associated cost in constructing seed alignments, but yields better performance over prior works.

## 2 Related Work

### 2.1 Entity Alignment

Until recently, entity alignment would require intensive human participation (Vrandečić and Krötzsch, 2014) to design hand-crafted features (Mahdisoltani et al., 2013), rules, or rely on external sources (Wang et al., 2017). In a broader context, works in schema and ontology matching also seek help from additional information by using e.g., extra data sources (Nguyen et al., 2011), entity descriptions (Lacoste-Julien et al., 2013; Yang et al., 2015), or semantics of the web ontology language (Hu et al., 2011). Performance of such schemes is bounded by the quality and availability of the extra information about the target KG, but obtaining sufficiently good-quality annotated data could be difficult for large KGs.

Recently, embedding-based entity alignment methods were proposed to reduce human involvement. JE (Hao et al., 2016) was among the first attempts in this direction. It learns embeddings of different KGs in a uniform vector space where entity alignment can be performed. MTransE (Chen et al., 2017) encodes KGs in independent embeddings and learns transformation between KGs. BootEA (Sun et al., 2018) exploits a bootstrapping process to learn KG embeddings. SEA (Pei et al., 2019) proposes a degree-aware KG embedding model to embed KGs. KDCoE (Chen et al., 2018) is a semi-supervised learning approach for co-training embeddings for multilingual KGs and entity descriptions. They all use translation-based models as the backbone to embed KGs.

Non-translational embedding-based methods include recent works on a GCN-based model (Wang et al., 2018b) and NTAM (Li et al., 2018). Additionally, most recent work, RDGCN (Wu et al., 2019), introduces the dual relation graph to model the relation information of KGs. Through

multiple rounds of interactions between the primal and dual graphs, RDGCN can effectively incorporate more complex relation information into entity representations and achieve promising results for entity alignment. However, existing methods only focus on entity embeddings and ignore the help that relation representations can provide on this task.

MTransE and NTAM are two of a few methods that try to perform both relation and entity alignments. However, both approaches require high-quality seed alignments, such as pre-aligned triples or relations, for relation alignment. Our approach advances prior works by jointly modeling entities and relations by using only a *small set* of pre-aligned entities (*but not relations*) to simultaneously perform entity and relation alignments.

## 2.2 Graph Convolutional Networks

GCNs (Duvenaud et al., 2015; Kearnes et al., 2016; Kipf and Welling, 2017) are neural networks operating on unlabeled graphs and inducing features of nodes based on the structures of their neighborhoods. Recently, GCNs have demonstrated promising performance in tasks like node classification (Kipf and Welling, 2017), relation extraction (Zhang et al., 2018a), semantic role labeling (Marcheggiani and Titov, 2017), etc. As an extension of GCNs, the R-GCNs (Schlichtkrull et al., 2018) have recently been proposed to model relational data for link prediction and entity classification. However, R-GCNs usually require a large number of parameters that are often hard to train, when applied to multi-relational graphs.

In this work, we choose to use GCNs to first encode KG entities and to approximate relation representations based on entity embeddings. Our work is the first to utilize GCNs for jointly aligning entities and relations for heterogeneous KGs.

## 3 Problem Formulation

We now introduce the notations used in this paper and define the scope of this work.

A KG is formalized 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 different KGs. Usually, some equivalent entities between KGs are already known, defined as *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 task of entity or relation align-

ment as automatically finding more equivalent entities or relations based on known alignment seeds. In our model, we only use known *aligned entity pairs* as training data for both entity and relation alignments. The process of relation alignment in our framework is unsupervised, which does not need pre-aligned relation pairs for training.

## 4 Our Approach

Given two target KGs,  $G_1$  and  $G_2$ , and a set of known aligned entity pairs  $\mathbb{L}$ , our approach uses GCNs (Kipf and Welling, 2017) with highway network (Srivastava et al., 2015) gates to embed entities of the two KGs and approximate relation semantics based on entity representations. By linking entity representations with relation representations, they promote each other in our framework and ultimately achieve better alignment results.

### 4.1 Overall Architecture

As illustrated in Figure 1, our approach consists of three stages: (1) preliminary entity alignment, (2) approximating relation representations, and (3) joint entity and relation alignment.

In the first stage, we utilize GCNs to embed entities of various KGs in a unified vector space for preliminary entity alignment. Next, we use the entity embeddings to approximate relation representations which can be used to align relations across KGs. In the third stage, we incorporate the relation representations into entity embeddings to obtain the joint entity representations, and continue using GCNs to iteratively integrate neighboring structural information to achieve better entity and relation representations.

### 4.2 Preliminary Entity Alignment

As shown in Figure 1, we put  $G_1$  and  $G_2$  in one graph  $G_a = (E_a, R_a, T_a)$  to form our model’s input. We utilize pre-aligned entity pairs to train our model and then discover latent aligned entities.

**Graph Convolutional Layers.** Our entity alignment model utilizes GCNs to embed entities in  $G_a$ . Our model consists of multiple stacked GCN layers so that it can incorporate higher degree neighborhoods. The input for GCN layer  $l$  is a node feature matrix,  $\mathbf{X}^{(l)} = \{\mathbf{x}_1^{(l)}, \mathbf{x}_2^{(l)}, \dots, \mathbf{x}_n^{(l)} | \mathbf{x}_i^{(l)} \in \mathbb{R}^{d^{(l)}}\}$ , where  $n$  is the number of nodes (entities) of  $G_a$ , and  $d^{(l)}$  is the number of features in layer  $l$ .  $\mathbf{X}^{(l)}$  is updated us-

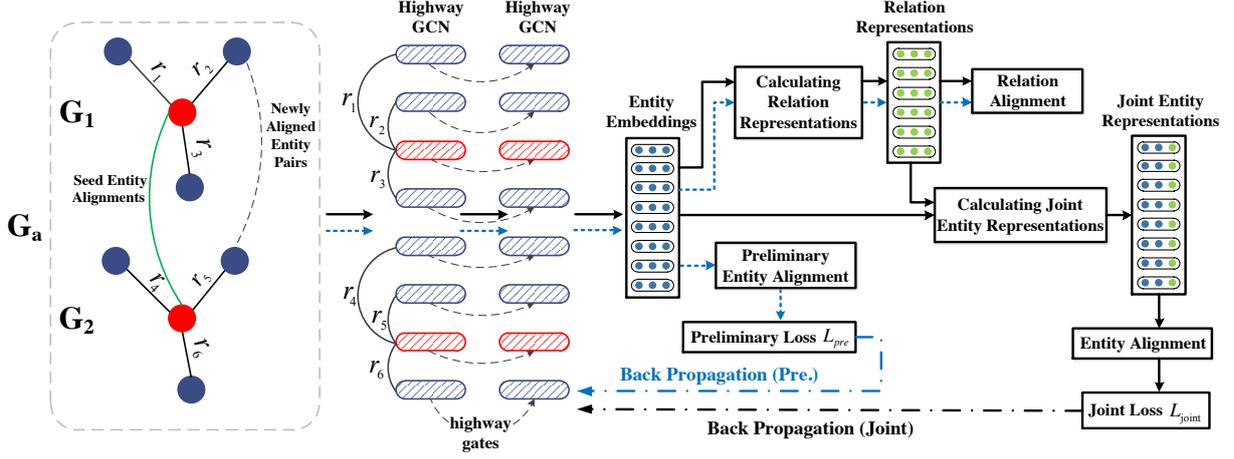


Figure 1: Overall architecture of our model. The blue dotted lines denote the process of preliminary entity alignment and preliminary relation alignment using approximate relation representations, and the black solid lines denote the process of continuing using GCNs to iteratively learn better entity and relation representations.

ing forward propagation as:

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

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

Inspired by (Rahimi et al., 2018) that uses highway gates (Srivastava et al., 2015) to control the noise propagation in GCNs for geographic localization, we also employ layer-wise highway gates to build a *Highway*-GCN (HGHCN) model. Our layer-wise gates work as follow:

$$T(\mathbf{X}^{(l)}) = \sigma(\mathbf{X}^{(l)} \mathbf{W}_T^{(l)} + \mathbf{b}_T^{(l)}), \quad (2)$$

$$\mathbf{X}^{(l+1)} = T(\mathbf{X}^{(l)}) \cdot \mathbf{X}^{(l+1)} + (\mathbf{1} - T(\mathbf{X}^{(l)})) \cdot \mathbf{X}^{(l)} \quad (3)$$

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

**Alignment.** In our work, entity alignment is performed by simply measuring the distance between two entity nodes on their embedding space. With the output entity representations  $\mathbf{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_n | \mathbf{x}'_i \in \mathbb{R}^d\}$ , for entities  $e_1$  from  $G_1$  and  $e_2$  from  $G_2$ , their distance is calculated as:

$$d(e_1, e_2) = \|\mathbf{x}'_{e_1} - \mathbf{x}'_{e_2}\|_{L_1}. \quad (4)$$

**Training.** We use a margin-based scoring function as the training objective, to make the distance

between aligned entity pairs to be as close as possible, and the distance between positive and negative alignment pairs to be as large as possible. The loss function is defined as:

$$L = \sum_{(p,q) \in \mathbb{L}} \sum_{(p',q') \in \mathbb{L}'} \max\{0, d(p, q) - d(p', q') + \gamma\}, \quad (5)$$

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

Rather than simply random sampling for negative instances, we look for more challenging negative samples, e.g., those with subtle differences from the positive ones, 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. 4 in the embedding space to replace  $q$  (or  $p$ ) as the negative instances.

### 4.3 Approximating Relation Representations

At this stage, we expect to obtain relation representations, which can be used in the next stage for constructing joint representations and can also be used for preliminary relation alignment. Since we are unable to explicitly modeling relations within our GCN-based framework, we thus approximate the relation representations based on their head and tail entity representations produced by the entity alignment model described in Section 4.2. This strategy is based on our observation that the statistical information of the head and tail entities of a relation can more or less reflect the shallow semantics of the relation itself, such as the head or tail entities' type requirements of a relation. Our

experiments in Section 6 suggest that this is a reasonable assumption.

Given a relation  $r \in R_a$ , there is a set of triples of  $r$ ,  $\mathbb{T}_r = \{(h_i, r, t_j) | h_i \in H_r, t_j \in T_r\}$ , where  $H_r$  and  $T_r$  are the sets of head entities and tail entities of relation  $r$ , respectively. For a relation  $r$ , its representation can be approximated as:

$$\mathbf{r} = f(\mathbf{H}_r, \mathbf{T}_r), \quad (6)$$

where  $\mathbf{r}$  is the approximated representation of relation  $r$ .  $\mathbf{H}_r$  and  $\mathbf{T}_r$  are the sets of HGCN-output embeddings of head entities and tail entities of relation  $r$ .  $f(\cdot)$  is a function to produce relation representations with input entity vectors, which can take many forms such as *mean*, *adding*, *concatenation* or more complex models. In our model, we compute the relation representation for  $r$  by first concatenating its averaged head and tail entity representations, and then introducing a matrix  $\mathbf{W}_R \in \mathbb{R}^{2\tilde{d} \times m}$  as a learnable shared linear transformation on relation vectors. Here,  $\tilde{d}$  is the number of features in each HGCN-output entity embedding and  $m$  is the number of features in each relation representation.

With the relation representations in place, relation alignment can be performed by measuring the similarity between two relation vectors. For relation  $r_1$  from  $G_1$  and  $r_2$  from  $G_2$ , their similarity is computed as:

$$s(r_1, r_2) = \|\mathbf{r}_1 - \mathbf{r}_2\|_{L_1} - \beta \frac{|P_{r_1 r_2}|}{|HT_{r_1} \cup HT_{r_2}|}, \quad (7)$$

where  $\mathbf{r}_1$  and  $\mathbf{r}_2$  are the relation representations for  $r_1$  and  $r_2$ . In addition to calculating the distance between the two relation vectors, we believe that the more aligned entities exist in the entities that are connected to the two relations, the more likely the two relations are equivalent. Thus, for  $r_1$  and  $r_2$ , we collect the pre-aligned entities existing in the head/tail entities of these two relations as the set  $P_{r_1 r_2} = \{(e_{i_1}, e_{i_2}) | e_{i_1} \in HT_{r_1}, e_{i_2} \in HT_{r_2}, (e_{i_1}, e_{i_2}) \in \mathbb{L}\}$ .  $HT_{r_1}$  and  $HT_{r_2}$  are the sets of head/tail entities for relation  $r_1$  and  $r_2$  respectively.  $\beta$  is a hyper-parameter for balance.

In our framework, relation alignment is explored in an unsupervised fashion, in which we do not have any pre-aligned relations as training data.

#### 4.4 Joint Entity and Relation Alignment

The first two stages of our approach could already produce a set of entity and relation alignments, but

we do not stop here. Instead, we attentively fuse the entity and relation representations and further jointly optimize them using the seed entity alignments. Our key insight is that entity and relation alignment tasks are inherently closely related. This is because aligned entities tend to have some relations in common, and similar relations should have similar categories of head and tail entities.

Specifically, we first pre-train the entity alignment model (Section 4.2) until its entity alignment performance has converged to be stable. We assume that both the pre-trained entity and approximate relation representations can provide rich information for themselves. Next, for each entity, we aggregate the representations of its relevant relations into a relation context vector, which is further combined with its pre-trained entity representation to form a new joint entity representation.

Formally, for each entity  $e \in E_a$ , its new joint representation  $e_{joint}$  can be calculated as:

$$e_{joint} = g(e, \mathbf{R}_e), \quad (8)$$

where  $e$  is the HGCN-output representation of entity  $e$ .  $\mathbf{R}_e$  is the set of relation representations of  $e$ 's relevant relations.  $g(\cdot)$  is a function to produce the new joint entity representation by taking  $e$  and  $\mathbf{R}_e$  as input, which can also take many forms of operations. In our model, we calculate  $e_{joint}$  by first summing all relation representations in  $\mathbf{R}_e$  and then concatenating  $e$  with the summed relation context vector.

After getting the new joint entity representations,  $\mathbf{X}_{joint}$ , we continue optimizing our model against the seed entity alignments, where we use the joint entity representations to calculate the training loss according to Eq. 5 to continue updating HGCNs<sup>1</sup>. Note that the joint entity representations are composed of entity embeddings and relation representations, while the relation representations are also constructed based on the entity embeddings. Hence, after backpropagation of the loss calculated using the joint entity representations, we optimize the entity embeddings.

## 5 Experimental Setup

### 5.1 Datasets

We use DBP15K datasets from (Sun et al., 2017) to evaluate our approach. DBP15K contains three

<sup>1</sup>The training procedure is detailed in Appendix A.

DBP15K		#Ent.	#Rel.	#Rel tr.	Alignments	
					#Ent.	#Rel.
ZH-EN	ZH	66,469	2,830	153,929	1,5000	890
	EN	98,125	2,317	237,674		
JA-EN	JA	65,744	2,043	164,373	1,5000	529
	EN	95,680	2,096	233,319		
FR-EN	FR	66,858	1,379	192,191	1,5000	212
	EN	105,889	2,209	278,590		

Table 1: Summary of the DBP15K datasets.

cross-lingual datasets that were built from the English version to Chinese, Japanese and French versions of DBpedia. Each contains data from two KGs in different languages and provides 15K pre-aligned entity pairs. Besides, each dataset also provides some pre-aligned relations. We manually aligned more relations from the three datasets and removed the ambiguously aligned relation pairs to construct the test sets for relation alignment. Table 1 shows the statistics of the three datasets. We stress that our approach achieves entity and relation alignments simultaneously using only a small number of pre-aligned entities, and relation alignments are only used for testing. Following the previous works (Sun et al., 2017; Wang et al., 2018b; Sun et al., 2018), we use 30% of the pre-aligned entity pairs as training data and 70% for testing. Our source code and datasets are freely available online<sup>2</sup>.

## 5.2 Implementation Details

We set  $\gamma = 1$ ,  $\beta = 20$ , and learning rate to 0.001. We sample  $\mathcal{K} = 125$  negative pairs every 50 epochs. We use entity names in different KGs for better model initialization. We translate non-English entity names to English via Google Translate, and the entity features are initialized with pre-trained English word vectors *glove.840B.300d*<sup>3</sup> in our model. Note that Google Translate does not always give accurate translations for named entities. We inspected 100 English translations for Japanese and Chinese entity names, and discovered that around 20% of the translations are wrong. The errors are mainly attributed to the missing of titles/modifications and wrong interpretations for person/location names. The inaccurate translation poses further challenges for our model.

## 5.3 Competitive Approaches

**Entity Alignment.** For entity alignment, we compare our approach against six embedding-based

<sup>2</sup><https://github.com/StephanieWyt/HGCN-JE-JR>

<sup>3</sup><http://nlp.stanford.edu/projects/glove/>

entity alignment methods discussed in Section 1: JE (Hao et al., 2016), MTransE (Chen et al., 2017), JAPE (Sun et al., 2017)<sup>4</sup>, IPTransE (Zhu et al., 2017), BootEA (Sun et al., 2018) and GCN (Wang et al., 2018b). Among those, BootEA is the best-performing model on DBP15K.

**Relation Alignment.** For relation alignment, we compare our approach with the state-of-the-art BootEA (denoted by BootEA-*R*), and MTransE (denoted by MTransE-*R*). Note that MTransE provides five implementation variants for its alignment model. To provide a fair comparison, we choose the one that does not use pre-aligned relations but gives the best performance for a triple-wise alignment verification (Chen et al., 2017) - a closely related task for relation alignment. Since BootEA and MTransE are translation-based models that encode both entities and relations, relation alignment can be done by measuring the similarities between two relation representations. Furthermore, to evaluate the effectiveness of our proposed relation approximation method, we also build BootEA-*PR* and MTransE-*PR* for relation alignment according to Section 4.3.

## 5.4 Evaluation Methodology

**Model Variants.** To evaluate our design choices, we provide different implementation variants with the following denotations. HGCN is our base GCN model with highway gates and entity name initialization. It has several variants, described as follows. HGCN-*PE* (Section 4.2) and HGCN-*PR* (Section 4.3) are our preliminary models for entity and relation alignments, respectively. HGCN-*JE* and HGCN-*JR* are our complete models that use joint representations to further improve entity alignment and relation alignment (Section 4.4). Finally, GCN-*PE* and GCN-*PR* are the preliminary GCN-based models for entity and relation alignments respectively, which use entity name initialization *but no highway gates*; GCN-*JE* and GCN-*JR* are the corresponding joint learning models; and GCN-*JE-r* is the randomly initialized version of GCN-*JE* without entity name initialization.

<sup>4</sup>We note that (Sun et al., 2017) also provides analysis by considering the outputs of a machine translator and JAPE, and using a theoretically perfect *oracle* predictor to correctly choose in between the results given by the machine translator and JAPE. As this only serves as an interesting up-bound analysis, but does not reflect the capability of JAPE (because it is impossible to build such a perfect predictor in the first place), we do not compare to this oracle implementation.

**Metrics.** Like prior works (Sun et al., 2017; Wang et al., 2018b; Sun et al., 2018), we use Hits@k as our evaluation metric. A Hits@k score is computed by measuring the proportion of correctly aligned entities ranked in the top  $k$  list. Hence, we prefer higher Hits@k scores that indicate better performance.

## 6 Experiment Results

In this section, we first show that our complete model consistently outperforms all alternative methods across datasets, metrics and alignment tasks. We then analyze the impact of prior alignment data size on model performance, showing that our approach requires significantly less training data but achieves better performance over the best-performing prior method. Finally, we use a concrete example to discuss how jointly learned entity and relation representations can be used to improve both entity and relation alignments.

### 6.1 Entity Alignment

Table 2 reports the performance for entity alignment of all compared approaches. The top part of the table shows the performance of prior approaches. By using a bootstrapping process to expand the training data, BootEA clearly outperforms all prior methods. By capturing the rich neighboring structural information, GCN outperforms all other translation-based models on Hits@1, and over IPTransE, MTransE and JE on Hits@10.

The bottom part of Table 2 shows how our proposed techniques, i.e., entity name initialization, joint embeddings and layer-wise highway gates, can be used within a GCN framework to improve entity alignment. After initialized with the machine-translated entity names, GCN-PE considerably improves GCN on all datasets. The improvement suggests that even rough translations of entity names (see Section 5.2) can still provide important evidence for entity alignment and finally boost the performance. By employing layer-wise highway gates, HGCN-PE further improves GCN-PE, giving a 34.31% improvement on Hits@1 on  $DBP15K_{FR-EN}$ , and also outperforms the strongest baseline BootEA. This substantial improvement indicates that highway gates can effectively control the propagation of noisy information. Our complete framework HGCN-JE gives the best performance across all metrics

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
<b>GCN-JE-r</b>	45.92	72.91	46.62	74.57	48.27	77.82
<b>GCN-PE</b>	50.70	79.23	53.01	82.83	54.42	84.65
<b>GCN-JE</b>	54.89	81.30	56.35	83.73	58.30	86.34
<b>HGCN-PE</b>	69.67	82.56	75.50	87.89	88.73	95.52
<b>HGCN-JE</b>	<b>72.03</b>	<b>85.70</b>	<b>76.62</b>	<b>89.73</b>	<b>89.16</b>	<b>96.11</b>

Table 2: Performance on entity alignment.

Models	ZH-EN		JA-EN		FR-EN	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
MTransE-R	3.03	8.88	2.65	10.21	3.30	14.62
MTransE-PR	32.81	57.64	31.00	56.14	18.87	44.34
BootEA-R	55.17	70.00	47.83	67.67	36.79	58.49
BootEA-PR	45.28	85.37	41.40	79.77	30.19	<b>60.38</b>
<b>GCN-PR</b>	66.18	82.81	60.87	81.47	38.21	52.83
<b>GCN-JR</b>	70.22	82.81	63.89	81.10	41.98	53.77
<b>HGCN-PR</b>	69.33	84.49	63.14	81.26	41.51	54.25
<b>HGCN-JR</b>	<b>70.34</b>	<b>85.39</b>	<b>65.03</b>	<b>83.55</b>	<b>42.45</b>	56.60

Table 3: Performance on relation alignment.

and datasets. Comparing HGCN-JE with HGCN-PE and GCN-JE with GCN-PE (2.36% and 4.19% improvements of Hits@1 on  $DBP15K_{ZH-EN}$  respectively), we observe that joining entity and relation alignments improves the model performance. Even without entity name initialization, GCN-JE-r still has obvious advantages over JE, MTransE, JAPE, IPTransE and GCN. The results reinforce our claim that merging the relation information into entities can produce better entity representations. We stress that our proposed methods are not restricted to GCNs or HGCNs, but can be flexibly integrated with other KG representation models as well.

### 6.2 Relation Alignment

Table 3 reports the results of relation alignment. Directly using the relation embeddings learned by MTransE to perform relation alignment leads to rather poor performance for MTransE-R, less than 4% for Hits@1 for all datasets. This is because the translation assumption,  $head + relation \approx tail$ , used by MTransE focuses on modeling the overall relationship among heads, tails, and relations, but capturing little neighboring information and relation semantics. After approximating the relation representations using entity embeddings according to Eq 6, MTransE-PR substantially improves MTransE-R. This confirms our assumption that it is feasible to approximate a relation using the in-

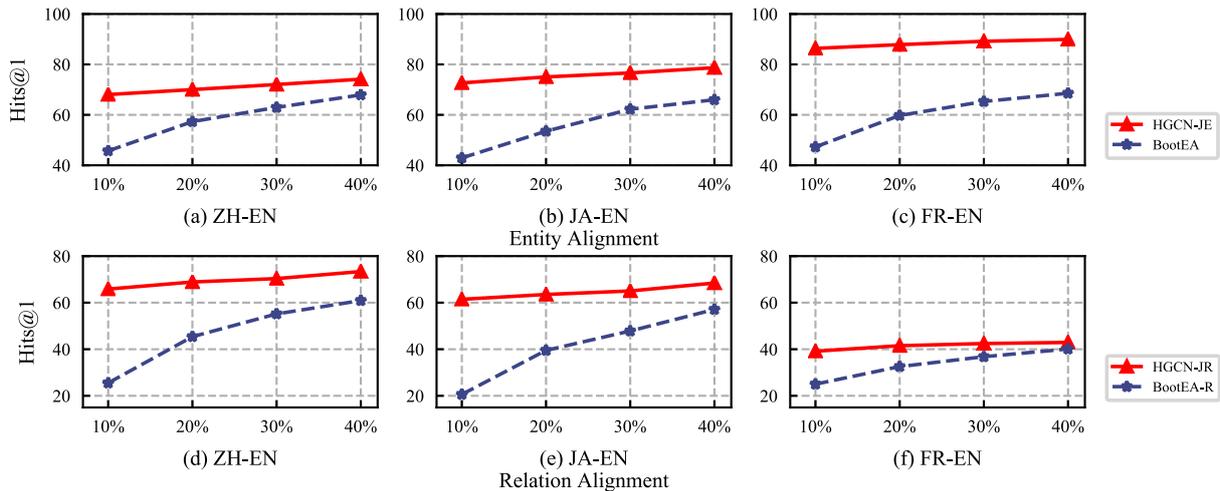


Figure 2: (a)-(c) report the performance for entity alignment of HGCN-JE and BootEA when they are trained with different proportions of seed entity alignments on the three DBP15K datasets. (d)-(f) show the relation alignment performance of HGCN-JR and BootEA-R under corresponding conditions. The x-axes are the proportions of seed alignments, and the y-axes are Hits@1 scores.

formation of its head and tail entities.

The strong entity alignment model BootEA also performs well for relation alignment. Using the relation embeddings from BootEA, BootEA-R delivers the best Hits@1 in MTransE and BootEA variants. Using our approximation strategy hurts BootEA-R in Hits@1, but we see improvements on Hits@10 across all datasets. This suggests that our approximation method can bring more related candidates, but may lack precision to select top-ranked candidates, comparing to explicitly relation modeling in translation-based models.

Our framework, HGCN-JR, delivers the best relation alignment results across datasets and metrics, except for Hits@10 on  $DBP15K_{FR-EN}$ . Like entity alignment, we also observe that joining entity and relation alignments improves relation alignment, as evidenced by the better performance of HGCN-JR and GCN-JR over HGCN-PR and GCN-PR, respectively. That is, joint modeling produces better entity representations, which in turn provide better relation approximations. This can promote the results of both alignment tasks.

### 6.3 Analysis

**Impact of Available Seed Alignments.** To explore the impact of the size of seed alignments on our model, we compare our HGCN with BootEA by varying the proportion of pre-aligned entities from 10% to 40% with a step of 10%. Figure 2 (a-c) illustrate the Hits@1 for entity alignment of HGCN-JE and BootEA on three datasets. As the

amount of seed alignments increases, the performances of both models on all three data sets gradually improve. HGCN-JE consistently obtains superior results compared to BootEA, and seems to be insensitive to the proportion of seed alignments. For example, HGCN-JE still achieves 86.40% for Hits@1 on  $DBP15K_{FR-EN}$  when only using 10% of training data. This Hits@1 score is 17.84% higher than that of BootEA when BootEA uses 40% of seed alignments.

Figure 2 (d-f) show the Hits@1 for relation alignment of HGCN-JR and BootEA-R. HGCN-JR also consistently outperforms BootEA-R, and gives more stable results with different ratios of seed entity alignments. These results further confirm the robustness of our model, especially with limited seed entity alignments.

**Case Study.** Figure 3 shows an example from  $DBP15K_{FR-EN}$ . In the stages of preliminary entity alignment and relation alignment, our model correctly predicts the aligned entity pair ( $v_2, v_5$ ) and relation pair ( $r_2, r_5$ ). After examining the full experimental data<sup>5</sup>, we find that the entities with more neighbors, such as  $v_2$  and  $v_5$  (indicating *Norway*), and the high-frequency relations, such as  $r_2$  and  $r_5$  (indicating *country*), are easier to align, since such entities and relations have rich structural information that can be exploited by a GCN. After jointly learning entity and relation representations, the extra neighboring relation information

<sup>5</sup>A more detailed analysis of our experimental results can be found in Appendix B in the supplementary material.

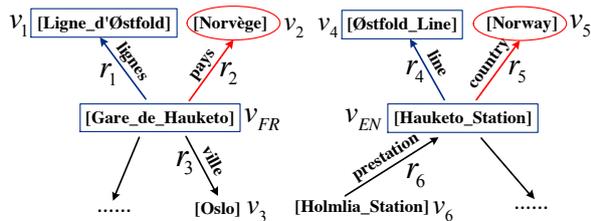


Figure 3: A real-world example from  $DBP15K_{FR-EN}$ .  $[v_2; v_5]$  and  $[r_2; r_5]$  are respectively the aligned entities and aligned relations after performing preliminary entity and relation alignments.  $[v_{FR}; v_{EN}]$  and  $[v_1; v_4]$  are the newly aligned entity pairs, and  $r_1$  and  $r_4$  are the newly aligned relations, which are discovered using jointly learned entity and relation representations. Jointly optimizing alignment tasks leads to the successful discovery of new aligned relation and entity pairs.

(e.g., the aligned relations ( $r_2, r_5$ )) enables our model to successfully align  $v_{FR}$  and  $v_{EN}$ . If we keep updating the model to learn better entity and relation representations, our alignment framework can successfully uncover more entity and relation alignments such as  $(v_1, v_4)$  and  $(r_1, r_4)$ . This shows that joint representations can improve both entity and relation alignments.

## 7 Conclusions

This paper presents a novel framework for entity alignment by jointly modeling entities and relations of KGs. Our approach does not require pre-aligned relations as training data, yet it can simultaneously align entities and relations of heterogeneous KGs. We achieve this by employing gated GCNs to automatically learn high-quality entity and relation representations. As a departure from prior work, our approach constructs joint entity representations that contain both relation information and entity information. We demonstrate that *the whole is greater than the sum of its parts*, as the joint representations allow our model to iteratively improve the learned representations for both entities and relations. Extensive experiments on three real-world datasets show that our approach delivers better and more robust performance when compared to state-of-the-art methods.

## Acknowledgments

This work is supported in part by the National Hi-Tech R&D Program of China (No. 2018YFB1005100), the NSFC Grants (No. 61672057, 61672058, 61872294), and a UK

Royal Society International Collaboration Grant (IE161012). For any correspondence, please contact Yansong Feng.

## References

- Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.
- Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*.
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.
- David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28*, pages 2224–2232.
- Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jian Zhao. 2016. A joint embedding method for entity alignment of knowledge bases. In *Proceedings of China Conference on Knowledge Graph and Semantic Computing (CCKS2016)*, pages 3–14.
- Wei Hu, Jianfeng Chen, and Yuzhong Qu. 2011. A self-training approach for resolving object coreference on the semantic web. In *Proceedings of the 20th international conference on World wide web*.
- Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):1–14.
- Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.
- Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. Sigma: Simple greedy matching for aligning large knowledge bases. In *Proceedings*

- of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.
- Shengnan Li, Xin Li, Rui Ye, Mingzhong Wang, Haiping Su, and Yingzi Ou. 2018. Non-translational alignment for multi-relational networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4180–4186.
- Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.
- Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515.
- Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1105–1116.
- Thanh Nguyen, Viviane Moreira, Huong Nguyen, Hoa Nguyen, and Juliana Freire. 2011. Multilingual schema matching for wikipedia infoboxes. *Proc. VLDB Endow*.
- Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference, WWW '19*, pages 3130–3136.
- Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2009–2019.
- Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. Paris: Probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168.
- Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644.
- Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4396–4402.
- Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57:78–85.
- Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. 2018a. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2246–2255.
- Xuepeng Wang, Kang Liu, Shizhu He, Shulin Liu, Yuanzhe Zhang, and Jun Zhao. 2017. Multi-source knowledge bases entity alignment by leveraging semantic tags. *Chinese Journal of Computers*, 40(3):701–711.
- Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018b. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357.
- Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5278–5284.
- Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.
- Yang Yang, Yizhou Sun, Jie Tang, Bo Ma, and Juanzi Li. 2015. Entity matching across heterogeneous sources. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018a. Graph convolution over pruned dependency trees improves relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018b. Variational reasoning for question answering with knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4258–4264.