

# Semi-supervised Entity Alignment via Joint Knowledge Embedding Model and Cross-graph Model

Chengjiang Li<sup>1,2,3</sup>, Yixin Cao<sup>4\*</sup>, Lei Hou<sup>1,2,3</sup>, Jiaxin Shi<sup>1,2,3</sup>, Juanzi Li<sup>1,2,3</sup>, Tat-Seng Chua<sup>4</sup>

<sup>1</sup>Department of Computer Science and Technology, Tsinghua University

<sup>2</sup>KIRC, Institute for Artificial Intelligence, Tsinghua University

<sup>3</sup>Beijing National Research Center for Information Science and Technology

<sup>4</sup>School of Computing, National University of Singapore, Singapore

{iamlockelighting, caoyixin2011, greener2009, shijx12}@gmail.com  
lijuanzi2008@gmail.com, dcscts@nus.edu.sg

## Abstract

Entity alignment aims at integrating complementary knowledge graphs (KGs) from different sources or languages, which may benefit many knowledge-driven applications. It is challenging due to the heterogeneity of KGs and limited seed alignments. In this paper, we propose a semi-supervised entity alignment method by joint Knowledge Embedding model and Cross-Graph model (KECG). It can make better use of seed alignments to propagate over the entire graphs with KG-based constraints. Specifically, as for the knowledge embedding model, we utilize TransE to implicitly complete two KGs towards consistency and learn relational constraints between entities. As for the cross-graph model, we extend Graph Attention Network (GAT) with projection constraint to robustly encode graphs, and two KGs share the same GAT to transfer structural knowledge as well as to ignore unimportant neighbors for alignment via attention mechanism. Results on publicly available datasets as well as further analysis demonstrate the effectiveness of KECG. Our codes can be found in <https://github.com/THU-KEG/KECG>.

## 1 Introduction

Recently, many Knowledge Graphs (KGs) (e.g., DBpedia (Lehmann et al., 2015), YAGO (Rebele et al., 2016) and BabelNet (Navigli and Ponzetto, 2012)) have emerged to provide structural knowledge for different applications. These separately constructed KGs contain heterogeneous but complementary contents; thus integrating KGs from different sources or languages into a unified KG becomes essential to better benefit knowledge-driven applications, ranging from information extraction (Han et al., 2018; Cao et al., 2018) to question answering (Cui et al., 2017).

\* Corresponding author

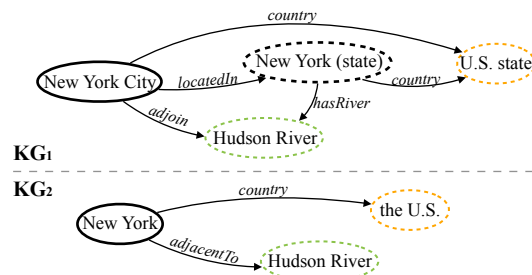


Figure 1: Example of the *New York City* entity and its neighbors in two heterogeneous KGs (Dashed circles in the same color means prior alignments).

Some efforts have been made to integrate KGs by aligning entities with their semantically same counterparts, namely entity alignment. Early entity alignment approaches either rely on human efforts (Mahdisoltani et al., 2013) or extra resources (Hu et al., 2011; Wang et al., 2013) to discover new equivalent entity pairs. Recent embedding approaches show significant improvement on entity alignment. We roughly classify these embedding models into two groups: KG-based models and graph-based models. KG-based models (Hao et al., 2016; Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Sun et al., 2018; Chen et al., 2018) utilize existing KG representation learning methods to learn embeddings of entities and relations in different KGs, and then align them into a unified vector space. This type of method can not only preserve KG structures, but also implicitly complete KG with the missing links from existing knowledge. However, KG-based methods require a sufficient number of seed alignments, which is usually expensive to obtain.

To alleviate the burden of seed alignments, graph-based methods (Wang et al., 2018) utilize Graph Convolutional Network (GCN) (Kipf and Welling, 2017) to enhance entity embeddings with their neighbors' information, thus can make better use of seed alignments to propagate them over

the entire graph. However, GCN-based models are sensitive to structural differences of KGs (Vaswani et al., 2017) and may not perform well for entity alignment, since different KGs are very heterogeneous. Another issue of this type of method is that they fail to consider relation types, which actually play an important role in entity alignment. As shown in Figure 1, the *New York City* entity has different surface forms in the two KGs (i.e., *New York City* in  $KG_1$  and *New York* in  $KG_2$ ). There are two possible mappings from the *New York City* and *New York (state)* in  $KG_1$  to the *New York* in  $KG_2$ , and the *New York* in  $KG_2$  shares the same neighbors *Hudson River* and *U.S. state* with *New York City* and *New York (state)* in  $KG_1$ . In this case, we need to further consider the relation types: *adjoin* and *hasRiver* to distinguish the two candidate alignments; otherwise, a graph-based method may be misled here.

In this paper, we propose a semi-supervised entity alignment method by joint Knowledge Embedding model and Cross-Graph model (KECG), which combines the above two types of methods. The basic idea of KECG is to utilize a cross-graph model to embed entities into a unified vector space by using inner-graph structure and inter-graph alignments information; meanwhile the knowledge embedding model learns KG representations to implicitly complete different KGs towards consistency and model relational constraints among entities. There are two key points for high-quality jointly training. First, the completion from KG learning may exacerbate the heterogeneity between two KGs, because two KGs may contain different rich parts, which shall become richer during training. Second, different from conventional KG representation learning, entity alignment requires one-to-one mapping, which implies that the similar entities sharing common neighbors cannot be embedded closely, otherwise they shall be aligned incorrectly as shown in Figure 1.

Therefore, as for cross-graph model, we propose to utilize an extended Graph Attention Network (GAT) (Vaswani et al., 2017) as our cross-graph model that allows robustly encoding of KGs. To alleviate the negative impacts of heterogeneity, we encode different KGs with the same GAT, so that it not only highlights the entities that are important for alignment in the graph, but also transfers structural knowledge via shared param-

eters. We also extend it with constraints on weight matrices, making our model more efficient and effective. Moreover, we employ the nearest neighbor sampling strategy to differentiate the entities that are embedded closely but not the same to further boost the performance. We summarize the main contributions as follows:

1. We propose a novel semi-supervised KECG model for entity alignment by joint knowledge embedding model and cross-graph model.
2. We utilize an extended GAT to alleviate the negative impacts of heterogeneous KGs, and employ the nearest neighbor sampling strategy for KG representation learning towards one-to-one mapping, making the combination more suitable for entity alignment.
3. We evaluate KECG on five publicly available datasets including three cross-lingual datasets and two large-scale datasets from different sources. Compared with four state-of-the-art baselines, experimental results and further analysis demonstrate the outperformance of KECG.

## 2 Background

### 2.1 Problem Formulation

Formally, we represent two heterogeneous KGs as  $G_1 = (E_1, R_1, T_1)$  and  $G_2 = (E_2, R_2, T_2)$ , where  $E_i, R_i, T_i$  represent the sets of entity, relation and triplet in  $G_i, i \in \{1, 2\}$ , respectively.  $N_e = \{e' | (e, r, e') \in T\} \cup \{e' | (e', r, e) \in T\}$  is the set of neighbors of the entity  $e$  in  $G$ .  $S = \{(e_1, e_2) \in E_1 \times E_2 | e_1 \leftrightarrow e_2\}$  is a set of labeled entity pairs that are same in semantics, where an equivalence relation  $\leftrightarrow$  holds between  $e_1$  and  $e_2$ , e.g.,  $e_1$  in  $G_1$  describes the same real-world entity with  $e_2$  in  $G_2$ . Entity alignment aims to find the remaining semantically same entity pairs in  $E_1 \times E_2$ . For convenience, we put  $G_1$  and  $G_2$  together into one large graph  $G = G_1 + G_2$ ,  $R = R_1 \cup R_2$ ,  $T = T_1 \cup T_2$ , and  $n = |E_1| + |E_2|$  is the total number of entities. We use bold-face letters to denote the vector representations of the corresponding terms throughout the paper.

### 2.2 Graph Neural Networks (GNNs)

GNNs are a type of neural network model that directly deals with structured data (Bruna et al.,

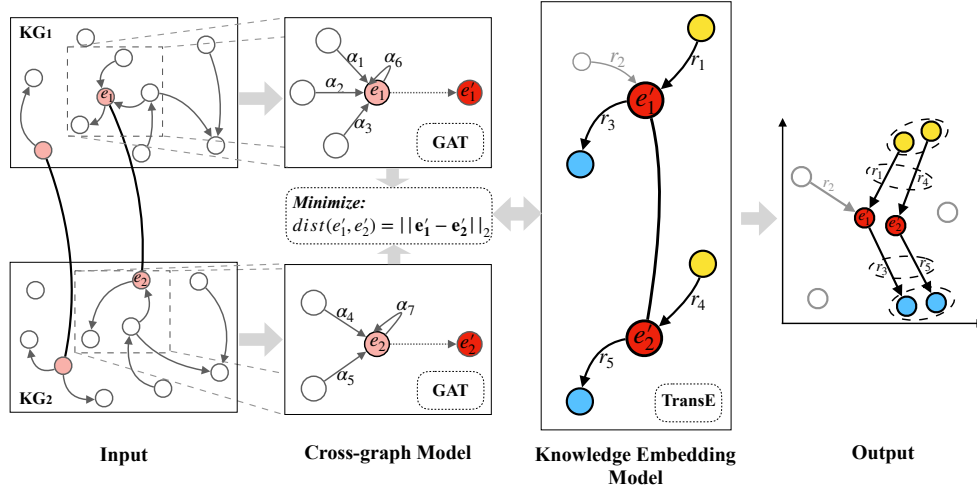


Figure 2: The overview framework of KECG. Solid lines (in Input) connect pre-aligned entities in two KGs. Nodes in different sizes (in Knowledge Embedding Model) show the attention of the center nodes over their neighbors. Circled nodes in the output are the potentially equivalent entities that we want to find out (in Output).

2014; Defferrard et al., 2016; Hamilton et al., 2017). They take a graph as an input, output labels for each node, and are similar to a propagation model: to enhance the features of a node according to its neighbor nodes. Among them, GCN (Kipf and Welling, 2017) is a simplification of spectral graph convolutions, whose formulation is as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{A}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

where  $\mathbf{A}$  is a normalized adjacent matrix of the input graph with self-connection,  $\mathbf{H}^{(l)}$  and  $\mathbf{W}^{(l)}$  are the hidden states and weights in the  $l_{th}$  layer, and  $\sigma(\cdot)$  is a non-linear activation, such as *ReLU*. Subsequently, Vaswani et al. (2017) introduce attention mechanism into GCN by dynamically calculating  $\mathbf{A}$  according to the attention values between each entity pairs, which is capable of mining structural knowledge better.

### 3 Method

To combine KG-based methods and graph-based methods with consideration of seed limit and KG heterogeneity, we propose a novel semi-supervised method KECG for entity alignment. As shown in Figure 2, KECG consists of two parts: cross-graph model and knowledge embedding model. The input of KECG includes a composition  $G$  of two KGs and their prior alignment set  $S$ .

#### 3.1 Cross-graph Model

The goal of our cross-graph model is to utilize structural information, including inner-graph

structure and inter-graph alignments, to embed entities into a unified vector space. Due to the incompleteness of KGs, there may be some entities and relations existing in one KG but missing in another KG. We utilize an extended GAT with projection constraint as an encoder to embed the KGs while paying different attention over their neighbors, therefore it is able to alleviate the negative impact of heterogeneity by ignoring some unimportant neighbors for alignment. Besides, the graph convolution operation allows us to leverage both the labeled entities and the abundant unlabeled entity information for alignment, making our method naturally semi-supervised.

The input of the encoder includes an entity embedding matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$  and neighbor sets of all entities as KG structure, where  $d$  is the dimension of entities. The KG encoder is built by stacking multiple GAT layers:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{A}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \quad (1)$$

where  $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times d^{(l)}}$  and  $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$  are the hidden states and weights in the  $l_{th}$  layer,  $\mathbf{H}^{(0)} = \mathbf{X}$ ,  $\sigma(\cdot)$  is a non-linear activation chosen as  $ReLU(\cdot) = \max(0, \cdot)$ , and  $\mathbf{A}^{(l)} \in \mathbb{R}^{n \times n}$  is a connectivity matrix computed by self-attention over the input graph. Considering a single element  $a_{ij}^{(l)}$  of  $\mathbf{A}^{(l)}$ , the weight from entity  $e_i$  to entity  $e_j$ , we compute it using self-attention mechanism:

$$a_{ij}^{(l)} = \text{softmax}(c_{ij}^{(l)}) = \frac{\exp(c_{ij}^{(l)})}{\sum_{e_k \in N_{e_i} \cup \{e_i\}} \exp(c_{ik}^{(l)})} \quad (2)$$

, where  $N_{e_i} \cup \{e_i\}$  is the neighbor set of  $e_i$  with self-loop,  $c_{ij}^{(l)}$  is the attention coefficient of entity  $e_i$  to entity  $e_j$ . Following Velickovic et al. (2018), the attention coefficient  $c_{ij}^{(l)}$  is computed as follows:

$$c_{ij}^{(l)} = \text{LeakyReLU}(\mathbf{q}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]) \quad (3)$$

where  $\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)} \in \mathbf{H}^{(l)}$  are the hidden states of  $e_i$  and  $e_j$ , respectively, *LeakyReLU* (Maas et al., 2013) is a nonlinear function,  $\mathbf{q} \in \mathbb{R}^{2d^{(l)}}$  is a learnable parameter,  $\cdot^T$  represents transposition and  $\oplus$  indicates vector concatenation. Let  $L$  be the number of layers. After  $L$  iterations of convolution, the hidden state  $\mathbf{H}^{(L)}$  integrates both features of entities and their neighbors, the  $i_{th}$  row of which represents the **attention enhanced embedding** of entity  $e_i$ .

**Projection Constraint.** Note that in every forward propagation, we use  $\mathbf{W}^{(l)}$  to transform from  $d^{(l)}$  dimension vector space to  $d^{(l+1)}$  dimension. Motivated by (Yang et al., 2015), we keep the dimensions of the layers and embeddings to be the same, and restrict the projection matrix  $\mathbf{W}$  to be a diagonal matrix. Such constraint can reduce the number of parameters and computations, increase the generalizability of the model. That’s why we claim that we extend the normal GAT.

**Objective  $O_C$ .** For semi-supervised entity alignment, we then minimize the representation distance of equivalent entities over all labeled alignments similar to (Wang et al., 2018):

$$O_C = \sum_{(e_i, e_j) \in S} \sum_{(e'_i, e'_j) \in S'} [dist(e_i, e_j) + \gamma_1 - dist(e'_i, e'_j)]_+ \quad (4)$$

where  $dist(e_i, e_j) = \|\mathbf{e}_i - \mathbf{e}_j\|_2$  is the  $L_2$  distance between the aligned entity pair  $(e_i, e_j)$ , representations of  $e_i$  and  $e_j$  are from the attention enhanced embedding matrix  $\mathbf{H}^{(L)}$ ,  $S'$  represents the negative pair set of  $S$  generated by nearest neighbor sampling (Kotnis and Nastase, 2017), and  $\gamma_1 > 0$  is a margin hyper-parameter. Because an entity can only have one corresponding entity in another KG, the entity closest to the corresponding entity in the same KG should be the best choice as a negative example to accurately discriminate the target entity. More formally, given a pre-aligned entity pair  $(e_i, e_j) \in S$ , where  $e_i \in E_1, e_j \in E_2$ ,  $K_2$  is the number of negative samples, we choose  $K_2$  entities that are nearest to  $e_j$  in  $E_2$  as the negative samples of  $e_i$ , and vice versa for  $e_j$ . After that,

each pre-aligned entity pair will have  $2 * K_2$  negative samples. We utilize the  $L_2$  distance as the measure to search for the nearest negative samples due to its superior performance in experiments.

### 3.2 Knowledge Embedding Model

The goal of our knowledge embedding model is to model inner-graph relationships, making entities more distinguishable. Here, we use TransE (Bordes et al., 2013), which is one of the most representative translation-based methods, as our knowledge embedding model. It is worth mentioning that other advanced KG learning methods can also be applied to our knowledge embedding model such as TransD (Ji et al., 2015), which is left for future work as our main idea is to joint cross-graph embeddings and knowledge embeddings for entity alignment.

**Objective  $O_K$ .** Formally, given a relational triplet  $(e_h, r, e_t)$ , TransE wants  $\mathbf{e}_h + \mathbf{r} \simeq \mathbf{e}_t$ . So it defines the score function  $f(e_h, r, e_t) = \|\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t\|_2$  to measure the plausibility of  $(e_h, r, e_t)$ , where  $\|\cdot\|_2$  means 2-norm. Following TransE, we utilize a margin-based ranking loss function as the training objective of the knowledge embedding model, defined as:

$$O_K = \sum_{(e_h, r, e_t) \in T} \sum_{(e'_h, r', e'_t) \in T'} [f(e_h, r, e_t) + \gamma_2 - f(e'_h, r', e'_t)]_+ \quad (5)$$

where  $[\cdot]_+ = \max\{0, \cdot\}$  represents the maximum between 0 and the input, embeddings of entities  $e_h$  and  $e_t$  are from the attention enhanced embedding matrix  $\mathbf{H}^{(L)}$ , relation  $r$  is from the relation matrix  $\mathbf{R} \in \mathbb{R}^{|R| \times d}$  which needs to be learned,  $T'$  stands for the negative sample triplet set of  $T$ , and  $\gamma_2 > 0$  is a margin hyper-parameter separating positive and negative triplets.  $T'$  is generated by corrupting  $T$ . For a triplet  $(e_h, r, e_t) \in T$ , we replace  $e_h$  to generate a negative triplet  $(e'_h, r, e_t)$ , where  $e'_h$  also has the same relation type  $r$  connect to other entities, and so does  $e_t$ .  $K_1$  is the number of negative samples for each positive triplet. For example, we think *Beijing City* can be a replacer for *New York City* in (*New York City, country, United States*), since *Beijing City* also has the relation *country* but connected with a different tail entity *China*. If the number of replacers is less than  $K_1$ , we randomly sample replacers from all entities as what the uniform negative sampling method does.



### 3.3 Optimization and Inference

**Overall Objective  $O$ .** Here, we define the objective function of KECG corresponding to the above two parts:

$$O = O_C + O_K \quad (6)$$

where  $O_C$  and  $O_K$  denote the objective function of the cross-graph model and the knowledge embedding model, respectively. It is worth noting that we can set different weights for  $O_C$  and  $O_K$ . But in order to treat both kinds of models equally, here, we set the same weight for them in experiments. We use AdaGrad (Duchi et al., 2011) to optimize the overall objective  $O$ .

**Inference.** When inferring, new equivalent entities are discovered based on the  $L_2$  distances between entities in the joint embeddings. Equivalent entities should become close with each other, so do equivalent relation types, e.g., circled nodes,  $r_1$  and  $r_4$  in the output of Figure 2.

## 4 Experiments

We evaluate KECG on entity alignment with five publicly available datasets including three cross-lingual datasets and two large-scale datasets from different sources. We compare KECG with four state-of-the-art baselines and conduct further analysis of essential components of KECG.

### 4.1 Datasets

Following (Sun et al., 2017, 2018; Wang et al., 2018), we use the DBP15K and the DWY100K datasets for evaluation. **DBP15K** contains three cross-lingual datasets built from DBpedia, denoted by  $DBP15K_{ZH-EN}$  (Chinese to English),  $DBP15K_{JA-EN}$  (Japanese to English) and  $DBP15K_{FR-EN}$  (French to English). Each dataset contains 15,000 reference entity alignments. To test the adaptability of KECG to large-scale data, we evaluate on **DWY100K**. **DWY100K** contains two large-scale datasets extracted from DBpedia, Wikidata and YAGO3, denoted by  $DWY100K_{WD}$  (DBpedia to Wikidata) and  $DWY100K_{YG}$  (DBpedia to YAGO3). Each dataset has 100,000 reference entity alignments. The detailed information of each dataset is listed in Table 1.

### 4.2 Experiment Settings

To investigate the ability of KECG, we compare it with the following state-of-the-art entity align-

Datasets	Entities	Relations	Rel. triplets	
DBP15K <sub>ZH-EN</sub>	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
DBP15K <sub>JA-EN</sub>	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
DBP15K <sub>FR-EN</sub>	French	66,858	1,379	192,191
	English	105,889	2,209	278,590
DWY100K <sub>WD</sub>	DBpedia	100,000	330	463,294
	Wikidata	100,000	220	448,774
DWY100K <sub>YG</sub>	DBpedia	100,000	302	428,952
	YAGO3	100,000	31	502,563

Table 1: Statistics of DBP15K and DWY100K.

ment methods including three KG-based models and one graph-based model.

- KG-based models: **MTransE** (Chen et al., 2017) is the basic KG-based method. **JAPE** (Sun et al., 2018) combines relation and attribute embeddings for entity alignment. **AlignEA** is the non-iterative version of BootEA (Sun et al., 2018) for entity alignment, which gets the best results.
- Graph-based model: **GCN-Align** (Wang et al., 2018) aligns cross-lingual entity via GCN.

More specifically, we use a 3-layer extended GAT as the encoder to process the composite KG  $G$  to generate embeddings of its entities. We set the dimension of all layers to be the same  $d$ . The entity embedding matrix  $\mathbf{X}$  and the relation matrix  $\mathbf{R}$  are initialized randomly and updated during training. Besides, we separated two variants from KECG for ablation study, called KECG(w/o NNS) and KECG(w/o K).

- KECG(w/o NNS) replaces the nearest neighbor sampling (NNS) strategy with normal uniform negative sampling.
- KECG(w/o K) simply removes the knowledge embedding model, only considering the equivalent relation as what GCN-Align does.

For all the compared approaches, we randomly split 30% of the reference entity alignments as prior alignments (i.e., training data) and leave the remaining as testing data. We use two evaluation metrics in this task: (1) the mean reciprocal rank of all correct entities (MRR) and (2) the proportion of correct entities that rank no larger than  $N$

Methods	DBP15K <sub>ZH-EN</sub>			DBP15K <sub>JA-EN</sub>			DBP15K <sub>FR-EN</sub>			DWY100K <sub>WD</sub>			DWY100K <sub>YG</sub>		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	30.83	61.41	0.364	27.86	57.45	0.349	24.41	55.55	0.335	28.12	51.95	0.363	25.15	49.29	0.334
JAPE	41.18	74.46	0.490	36.25	68.50	0.476	32.39	66.68	0.430	31.84	58.88	0.411	23.57	48.41	0.320
AlignEA	47.18	79.19	0.581	44.76	78.89	0.563	48.12	82.43	0.599	56.55	82.70	0.655	63.29	84.76	0.707
GCN-Align	41.25	74.38	0.549*	39.91	74.46	0.546*	37.29	74.49	0.532*	50.60*	77.21*	0.600*	59.72*	83.80*	0.682*
KECG(w/o NNS)	41.10	77.67	0.549	41.05	77.55	0.550	39.30	77.17	0.547	46.95	76.29	0.571	57.10	80.58	0.653
KECG(w/o K)	43.04	79.12	0.551	44.61	80.70	0.567	43.22	81.45	0.559	59.77	87.99	0.695	<b>73.83</b>	<b>93.04</b>	<b>0.806</b>
KECG	<b>47.77</b>	<b>83.50</b>	<b>0.598</b>	<b>48.97</b>	<b>84.40</b>	<b>0.610</b>	<b>48.64</b>	<b>85.06</b>	<b>0.610</b>	<b>63.23</b>	<b>89.97</b>	<b>0.726</b>	72.76	91.51	0.795

“\*\*” denotes the unreported metrics in their papers. We reproduced the results using their source code.

Table 2: Results comparison on entity alignment(Results of Hits@1 and Hits@10 are percentage values).

(Hits@N). Higher values of MRR and Hits@N indicate a better entity alignment model.

For parameter settings, the dimension of the hidden units in each GAT layer is set to be the same  $d = 128$  as the entity and relation embeddings. Learning rate  $\lambda$  for AdaGrad is chosen among  $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ . Margin  $\gamma_1$  and  $\gamma_2$  are chosen among  $\{1.0, 2.0, 3.0\}$ . The number of negative samples for cross-graph model and knowledge model are  $K_1 = 25$  and  $K_2 = 2$ , respectively, and the negative samples will be updated every 10 epochs. The total number of training rounds is 1000. The optimal configuration of our models for the entity alignment task is:  $\lambda = 0.005$ ,  $\gamma_1 = 3.0$ ,  $\gamma_2 = 3.0$ .

### 4.3 Results

Table 2 lists the results of entity alignment on DBP15K and DWY100K. In general, KECG is significantly more effective than MTransE, JAPE, and GCN-Align on all datasets, and slightly outperform AlignEA on small-scale datasets but greatly on large-scale datasets. Among all the baselines, AlignEA is the strongest one, which results from it using two margins to control the scores of triplets, making the KG structure modeling based on triplets more accurate than others. We conclude the detailed results in three aspects:

- **For different model groups.** Compared with the KG-based models, KECG performs much better than MTransE and JAPE. On DBP15K<sub>ZH-EN</sub>, KECG and AlignEA get very close results regarding Hits@1 with a gap of 0.59%, but as for Hits@10 and MRR, the differences become 4.31% and 0.017, respectively. The reason is that, though some entities are not accurately aligned, their neighbors, or neighbors of neighbors, may already be aligned. And KECG propagates these features to unaligned entities, making

the positions of the unaligned entities in the vector space very close to the corresponding entities. It reflects the importance of the global structural information of KGs. Compared with the graph-based model, KECG significantly outperforms GCN-Align regarding all metrics. It shows that involving the local relationships between entities can make entities more distinguishable, making the alignment more accurate.

- **For different data scales.** It can be seen that KECG has good adaptability on different data scales. The average gains of KECG regarding Hits@1 are 8%-41.3% on large-scale datasets and 1.7%-20.7% on small-scale datasets. With large-scale KGs providing richer relational and structural information, KECG can encode entities with more semantics. However, we found that, on DWY100K<sub>YG</sub>, KECG’s variant KECG(w/o K) works better than itself. This will be further discussed in Section 4.4.
- **For different language pairs.** KECG performs best on all language pairs. It improves the MRR score to 0.61 on DBP15K<sub>JA-EN</sub>, which is a nearly 8% improvement compared with the best baseline. To the best of our knowledge, the imbalanced resources of non-English KGs with English KG result in the heterogeneity of cross-lingual KGs, making it challenging to align them. KG-based models, such as MTransE, cannot learn good representations to model the structures of non-English KGs, as restraints for entities are relatively insufficient. Methods based on GCNs, such as GCN-Align, are sensitive to structural differences, and find it hard to narrow the differences for alignment. KECG makes use of the extended GAT to alleviate the neg-

ative impacts of heterogeneous KGs, which successfully highlights the entities that are important for alignment in the graph. Because English and French share many similarities, both KECG and baselines get satisfactory results on  $DBP15K_{FR-EN}$ .

## 4.4 Analysis

### 4.4.1 Effectiveness of Knowledge Embedding Model

For better comparison, we draw two detailed Hits@N bar plots of GCN-Align, AlignEA, KECG(w/o K) and KECG on  $DBP15K_{ZH-EN}$  and  $DWY100K_{YG}$ , as shown in Figure 3. In Figure 3(a), KECG(w/o K) gets better results than GCN-Align, which reflects the power of the graph attention mechanism. However, KECG(w/o K) cannot exceed AlignEA, showing that only considering structural information is not enough for entity alignment. After adding the knowledge embedding model, KECG achieves the highest Hits@{1,10,50}. The effectiveness of knowledge embedding model lies in two parts. Firstly, it applies constraints to entities by involving representation of relation types. Entity alignment leads to some relationship alignment, and entities having similar relations become close. Secondly, the knowledge embedding model is beneficial for the graph attention mechanism to learn better attention values.

In Figure 3(b), KECG still outperforms GCN-Align and AlignEA, but KECG(w/o K) exceeds KECG regarding three metrics, which reflects the weakness of the current knowledge embedding model. According to our statistics, the 1-to-N, N-to-1, and N-to-N relations account for 79.58% of the total in  $DWY100K_{YG}$ , while 60.05% in  $DBP15K_{ZH-EN}$ . The introduced knowledge embedding model is based on TransE, lacking the ability of modeling complex relationships (e.g., 1-to-N, N-to-1, and N-to-N relations). When there are too many complex relationships in the datasets, it brings more decrement rather than improvement to the results. To ease this situation, we can use a stronger representation model for complex relationships, such as TransR (Lin et al., 2015). We leave this for future work.

### 4.4.2 Effectiveness of Nearest Neighbor Sampling

In Table 2, it can be seen that KECG(w/o NNS) outperforms many baselines and its Hits@1 on

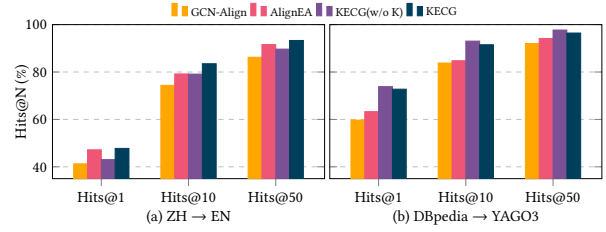


Figure 3: Detailed Hits@N for investigating the effectiveness of Knowledge Embedding Model.

$DBP15K_{FR-EN}$  reaches 39.30%, which is higher than another graph-based method GCN-Align by 2.01%. But the Hits@1 of KECG(w/o NNS) is lower than GCN-Align’s by approximately 3% on  $DWY100K$ . It is because  $DWY100K$  has a larger number of triplets, while the number of relations is relatively small. Involving the representation of complex relationships will introduce noise and affect the performance of the cross-graph model. After employing the nearest neighbor sampling strategy in cross-graph model, KECG considerably improves the results of KECG(w/o NNS), e.g., Hits@1 on  $DWY100K_{WD}$  is raised from 46.95% to 63.23%. The excellent performance of nearest neighbor sampling owing to its accurately discriminating the target entity from other close entities, especially on large-scale datasets.

### 4.4.3 Sensitivity to Proportion of Prior Alignment

To investigate whether KECG is sensitive to the proportion of pre-aligned entities, we vary the proportion from 10% to 50% with step 10% for training, and use the rest of pre-aligned entities for testing. Intuitively, more pre-aligned entities should provide more information to discover the potential pairs of equivalent entities.

Figure 4 illustrates the changes in Hits@1 and Hits@10 of KECG, AlignEA and GCN-Align on all datasets. As expected, the results of all methods become better with the increase in proportion. Given half of the prior alignment like 50%, all three methods get close and satisfactory Hits@1. When given a very small proportion of prior alignment like 10%, the results of AlignEA and GCN-Align decrease enormously, but our KECG still achieves outstanding results. Therefore, KECG works well on entity alignment even with a limited number of prior alignment.

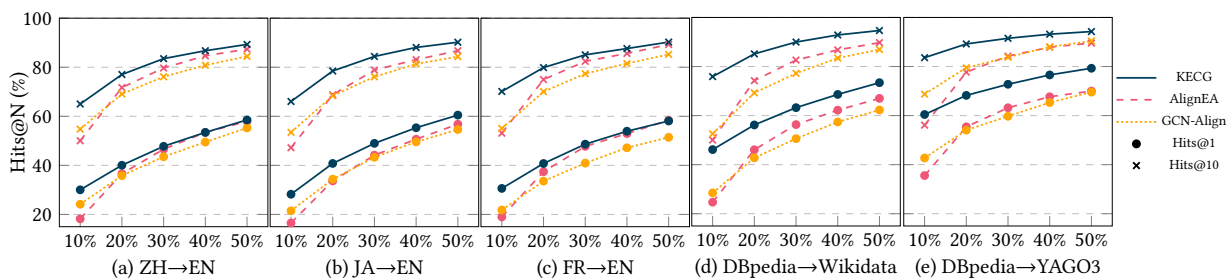


Figure 4: Hits@N of KECG, AlignEA and GCN-Align on entity alignment w.r.t. proportion of prior alignment.(The x-axis represents the proportion of prior alignment.)

## 5 Related Work

### 5.1 KG Embedding

Various KG embedding methods have shown effectiveness in modeling the semantic information of KGs. TransE (Bordes et al., 2013) is a milestone in learning embeddings for KGs, which interprets a relation as the translation from its head entity to its tail entity. Then it motivated several enhanced methods like TransR (Lin et al., 2015). In addition to them, non-translational methods also achieve satisfactory performances, such as RESCAL (Nickel et al., 2011), ConvE (Dettmers et al., 2018) and RotatE (Sun et al., 2019). Meanwhile, external information in KGs is fused to improve embedding (Wang and Li, 2016). More detailed KG embedding methods are summarized in (Wang et al., 2017).

### 5.2 Entity Alignment

Early human efforts, such as crowdsourcing and hand-crafted features (Lehmann et al., 2015; Mahdisoltani et al., 2013), are utilized to address the entity alignment problem. Though reaching high precisions, they are usually time-consuming and costly. Many automated approaches leverage the extra resources, such as OWL properties and Wikipedia links (Hu et al., 2011; Wang et al., 2013) to align entities, but such extra information is not generally available for all KGs.

Recently, KG embedding-based approaches become the most popular solution for entity alignment. JE (Hao et al., 2016) is an early work to jointly embed different KGs into a uniform vector space to align entities in KGs. MTransE (Chen et al., 2017) learns transitions to translate each entity embedding vector to its counterparts in another space. JAPE (Sun et al., 2017) jointly trains the attribute embedding and structure embedding by a skip-gram model and a translational model,

respectively, to align entities. GCN-Align (Wang et al., 2018) is a new graph-based model aligning KGs via GCN. It takes advantage of GCN to propagate information from neighbors, and align entity embeddings enhanced by structural knowledge. However, GCN-Align only considers the equivalent relations between entities in model training, neglecting the use of abundant relationships in KGs to precisely distinguish entities with similar neighbors. At the same time, iterative alignment becomes a new fashion to improve the results of entity alignment. IPTransE (Zhu et al., 2017) and BootEA (Sun et al., 2018) integrate knowledge among different KGs by enlarging the training data (prior alignments) in a bootstrapping way. KDCoE (Chen et al., 2018) iteratively co-trains multilingual KG embeddings and fuses them with entity description information for alignment. The aforementioned iterative methods improve the results mainly by increasing the number of pre-align entity pairs while training, and such a strategy can be a general enhancement for most alignment approaches (Sun et al., 2018). A better non-iterative method should achieve better results through bootstrap. Therefore, we are more concerned with the best result that a non-iterative method can achieve.

## 6 Conclusions

In this paper, we propose a semi-supervised entity alignment method KECG that combines the knowledge embedding model and graph-based model. We utilize an extended GAT to encode heterogeneous KGs and perform entity alignment by propagating alignment information over the entire graphs. Meanwhile, the KG learning is involved to model different relation types, making entities more distinguishable. Experimental results show that KECG significantly outperforms four state-of-the-art baselines.

For future work, we will extend the knowledge



embedding model of KECG to other KG representation learning methods, such as TransD (Ji et al., 2015), to gain a stronger ability of modeling relationships. Besides, iteratively discovering new entity alignments based on the framework of KECG is another interesting direction.

## Acknowledgments

This work is supported by National Key Research and Development Program of China (2017YFB1002101), NSFC key projects (U1736204, 6153301), a research fund by Alibaba, and THUNUS NEXT Co-Lab. NEXT++ research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its IRC@SG Funding Initiative.

## References

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.
- Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.
- Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *COLING*.
- 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 *IJCAI*.
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*.
- Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. Kbqa: learning question answering over qa corpora and knowledge bases. *PVLDB*.
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*.
- Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*.
- Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.
- Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *AAAI*.
- Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint embedding method for entity alignment of knowledge bases. In *CCKS*.
- Wei Hu, Jianfeng Chen, and Yuzhong Qu. 2011. A self-training approach for resolving object coreference on the semantic web. In *WWW*.
- Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*.
- Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.
- Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *ICML*.
- Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*.
- Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *ISWC*.
- Zejun Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*.

- Zequan Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *TKDE*.
- Zhichun Wang, Juanzi Li, and Jie Tang. 2013. Boosting cross-lingual knowledge linking via concept annotation. In *IJCAI*.
- Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*.
- Zhigang Wang and Juanzi Li. 2016. Text-enhanced representation learning for knowledge graph. In *IJCAI*.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.
- Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*.