

# Iterative Entity Alignment via Knowledge Embeddings

Hao Zhu<sup>1\*</sup>, Ruobing Xie<sup>1\*</sup>, Zhiyuan Liu<sup>1,2†</sup>, Maosong Sun<sup>1,2</sup>

<sup>1</sup> Department of Computer Science and Technology,  
State Key Lab on Intelligent Technology and Systems,  
National Lab for Information Science and Technology, Tsinghua University, Beijing, China  
<sup>2</sup> Jiangsu Collaborative Innovation Center for Language Ability,  
Jiangsu Normal University, Xuzhou 221009 China

## Abstract

Entity alignment aims to link entities and their counterparts among multiple knowledge graphs (KGs). Most existing methods typically rely on external information of entities such as Wikipedia links and require costly manual feature construction to complete alignment. In this paper, we present a novel approach for entity alignment via joint knowledge embeddings. Our method jointly encodes both entities and relations of various KGs into a unified low-dimensional semantic space according to a small seed set of aligned entities. During this process, we can align entities according to their semantic distance in this joint semantic space. More specifically, we present an iterative and parameter sharing method to improve alignment performance. Experiment results on real-world datasets show that, as compared to baselines, our method achieves significant improvements on entity alignment, and can further improve knowledge graph completion performance on various KGs with the favor of joint knowledge embeddings.

## 1 Introduction

Knowledge graphs (KG), aiming to organize human knowledge in structural forms, are playing an increasingly important role as infrastructural facilities of artificial intelligence and natural language processing. A typical KG is usually represented using triple facts of (*head entity, relation, tail entity*), also abridged as (*h, r, t*). Recent years have witnessed the vigorous development of large-scale KGs such as Freebase [Bollacker *et al.*, 2008], DBpedia [Lehmann *et al.*, 2015] and YAGO [Suchanek *et al.*, 2007]. These KGs provide well-structured information about entities and relations in the world and are widely used as prior knowledge in applications such as question answering [Yin *et al.*, 2016] and language modeling [Ahn *et al.*, 2016].

\*indicates equal contribution

†Corresponding author: Z. Liu (liuzy@tsinghua.edu.cn)

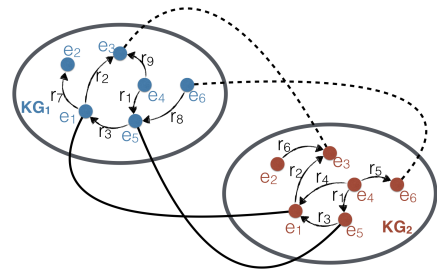


Figure 1: KG<sub>1</sub> and KG<sub>2</sub> denote two heterogeneous knowledge graphs. Solid lines connecting entities and their counterparts demonstrate alignment seeds, while dashed lines demonstrate counterpart relationship to be discovered.

Various methods, sources, and languages have been explored to construct KGs, and most existing KGs are developed separately. These KGs are inevitably heterogeneous in surface forms and typically supplementary in contents. It is thus essential to align entities in multiple KGs and join them into a unified KG for knowledge-driven applications.

Some efforts have been devoted to knowledge alignment. These conventional methods usually involve collaborative efforts [Vrandečić and Krötzsch, 2014] or focus on certain well-structured schemas or ontologies such as infobox in Wikipedia [Rinser *et al.*, 2013]. These methods incline to use external information of KGs, which are typically constrained to either scalability due to intensive human labor, or specific application scenarios with less flexibility.

Most existing works have not well studied the effectiveness of rich internal information in KGs for entity alignment. To better investigate this problem, we formalize entity alignment as that given a small seed set of aligned entities; we link entities and their counterparts among multiple heterogeneous KGs solely according to their internal structures. Fig. 1 demonstrates the overall architecture of entity alignment. Note that, since relations are universal and thus in this paper, we suppose all relations are already shared among various KGs.

The key challenge of this task is how to better model and utilize large-scale structural information of various KGs, which are always incomplete and even noisy. In this paper, we present to model internal structural information of KGs

using knowledge representation learning (KRL) and propose a novel approach for entity alignment with the favor of knowledge embeddings derived from KRL. We extend translation-based methods (e.g. TransE) [Bordes *et al.*, 2013], one of the most widely-used KRL approaches, to encode entities and relations of various KGs into a unified low-dimensional vector space, known as knowledge embeddings. In this semantic space, those entities of the identical or related meanings tend to be close to each other. Hence, the counterparts of an entity are expected to be close to this entity in the semantic space, and we can simply perform entity alignment between them.

More specifically, our method consists of three parts: (1) **Knowledge Embeddings.** We learn both entity and relation embeddings following the translation-based KRL methods according to the triple facts in various KGs. (2) **Joint Embeddings.** We learn to map knowledge embeddings of various KGs into a joint semantic space according to a seed set of known aligned entities. (3) **Iterative Alignment.** We iteratively align entities and their counterparts and update the joint knowledge embeddings by taking those high-confident aligned entities increasingly found by our method into consideration.

In experiments, we evaluate our method on two tasks including entity alignment and heterogeneous knowledge graph completion. The results show that our method can efficiently model internal structural information of KGs and discover entity alignments among various KGs. Moreover, the knowledge embeddings learned from heterogeneous KGs can significantly improve knowledge completion as compared to those learned from individual KGs.

## 2 Related Work

### 2.1 Knowledge Alignment

In general, conventional knowledge alignment usually involves heavy human collaborative efforts such as crowdsourcing [Vrandečić and Krötzsch, 2014] or well-designed hand-crafted features [Mahdisoltani *et al.*, 2014]. These works can achieve high alignment accuracies, while the human-involved approach is time-consuming, labor-expensive and usually suffers from extension inflexibility.

As for graph-based models, there are also many methods focusing on using heterogeneous information in different knowledge graphs for knowledge alignment. such as [Nguyen *et al.*, 2011; Wang *et al.*, 2013; Rinser *et al.*, 2013; Lacoste-Julien *et al.*, 2013]. [Niu *et al.*, 2012; Pershina *et al.*, 2015] improve the alignment with iterative methods. Though reaching high precisions, they are usually time-consuming on large-scale KGs.

As for embedding models, [Chen *et al.*, 2016] directly uses structural information in KG for multilingual knowledge alignment based on known aligned triple facts. Moreover, [Dong *et al.*, 2014] also inspires knowledge alignment with the collaboration of cross-modal information. Embedding methods are more suitable for large-scale knowledge graph, so we follow the fashion of embedding models.

Differing from these methods, we propose a novel embedding approach with iterative entity alignment, which only uses internal structural information of KGs for knowledge

representation learning. We further propose a soft alignment strategy marking possible entity alignments with probabilities, which could reconsider existing soft alignments during certain iterations.

### 2.2 Knowledge Representation Learning

TransE [Bordes *et al.*, 2013] projects both entities and relations into a continuous low-dimensional vector space, interpreting relations as translating operations between head and tail entities. TransE assumes that in the vector space we have  $\mathbf{h} + \mathbf{r} \simeq \mathbf{t}$ , which is simple and effective. However, TransE only considers individual triples of KGs for representation learning, regardless of multi-step relation paths. To address this issue, [Lin *et al.*, 2015] proposes PTransE to encode multi-step paths for KRL. There are existing some other methods to learn knowledge representations: RESCAL [Nickel *et al.*, 2011; 2012] and HOLE [Nickel *et al.*, 2016] are based on tensor factorization with relations considered to be matrices. NTN [Socher *et al.*, 2013] utilizes a neural layer of relation-specific tensors to jointly model head and tail embeddings. In this paper, we extend the most widely-used TransE and its effective extension PTransE as our KRL methods for entity alignment.

## 3 Problem Formulation

We first introduce the notations used in this paper. We describe knowledge in knowledge graph as triples  $(h, r, t)$ , in which  $h$  and  $t$  denote head and tail entities and  $r$  denotes the relations between entities. A knowledge graph is formalized as  $KG = (E, R, T)$ , where  $E, R, T$  are the set of entities, relations and triples respectively.

Suppose there are multiple knowledge graphs  $\Sigma = \{KG_i | KG_i = (E_i, R_i, T_i)\}$  of heterogeneous and complementary triples. That is, an entity in a KG has its counterparts in other KGs in different languages or surface names. We call these entities *synonymous entities*. In practice, some *synonymous entities* among KGs are already known, defined as *alignment seeds*  $\mathbb{L} = \{(e_{i_1}, e_{i_2}, \dots, e_{i_{|\Sigma|}}) | e_{i_j} \in E_j \wedge (e_{(i_1)_j} \neq e_{(i_2)_j}, \forall i_1 \neq i_2)\}$ . Each pair of entities from *alignment seeds* is also called *aligned entities*. The task of entity alignment is to automatically find and align more *synonymous entities* based on known alignment seeds. As compared to the number of entities, the set of relations is much smaller, which can be easily aligned manually or automatically. Hence, we could assume that all of the alignments between relations are known.

In this paper, we propose our method to solve the following issues: (1) aligning entities in various KGs and (2) jointly learning better knowledge embeddings with heterogeneous KGs.

## 4 Method

Without loss of generality, we introduce our method based on two heterogeneous KGs:  $KG_1 = (E_1, R_1, T_1)$  and  $KG_2 = (E_2, R_2, T_2)$  for entity alignment. Our method will take the triple facts in these KGs and *alignment seeds* as inputs to learn joint knowledge embeddings and align entities with their counterparts simultaneously.

## 4.1 Overall Architecture

As mentioned in Section 1, our method consists of three parts: knowledge embeddings, joint embeddings, and iterative alignment. Hence, we define the objective function of our method corresponding to the three parts:

$$L = \mathcal{K} + \mathcal{J} + \mathcal{I}, \quad (1)$$

where  $\mathcal{K}$ ,  $\mathcal{J}$  and  $\mathcal{I}$  denote the score function of knowledge embeddings, joint embeddings, and iterative alignment. Fig. 2 demonstrates the overall architecture of our model. In the following sections, we introduce the three parts in details.

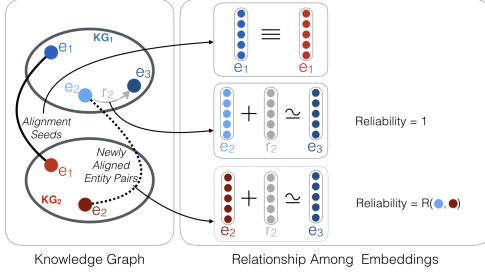


Figure 2: Overall architecture. This figure indicates our method implemented through TransE with Parameter Sharing and Soft Alignment. Blue and Red points denote entities from  $KG_1$  and  $KG_2$  respectively, while gray arrow denotes relations in both  $KG_1$  and  $KG_2$ . The Solid line and the dashed line between KGs denote *alignment seeds* and *newly aligned entity pairs* during iterative learning. We use links between KGs and score functions to indicate embedding sources and corresponding destinations. The same color represents the same entity/relation and corresponding embeddings.

## 4.2 Knowledge Embeddings

We obtain knowledge embeddings following the most widely-used translation-based method TransE and its effective extension PTransE. This part can also be implemented through other KRL methods which may be left as future work.

**TransE** TransE [Bordes *et al.*, 2013] projects both relations and entities into the same continuous low-dimension vector space, in which the relations are considered as translating vectors from head entities to tail entities. That is, TransE wants  $\mathbf{h} + \mathbf{r} \simeq \mathbf{t}$ . The energy function is defined as:

$$E(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|. \quad (2)$$

We utilize a margin-based score function as the training objective, defined as:

$$\mathcal{K}_T = \sum_{T \in \{T_1, T_2\}} \sum_{(h, r, t) \in T} L(h, r, t), \quad (3)$$

where  $L(h, r, t)$  is a margin-based loss function with respect to the triple  $(h, r, t)$ :

$$L(h, r, t) = \sum_{(h', r', t') \in T^-} [\gamma + E(h, r, t) - E(h', r', t')]_+, \quad (4)$$

where  $[x]_+ = \max\{0, x\}$  represents the maximum between 0 and  $x$ .  $T^-$  stands for the negative sample set of  $T$ , define

as follows:

$$T^- = \{(h', r, t) | h' \in E\} \cup \{(h, r, t') | t' \in E\} \cup \{(h, r', t) | r' \in R\}, \quad (h, r, t) \in T. \quad (5)$$

This indicates one of three components in a triple is randomly replaced by others.

**PTransE** TransE achieves significant performance in many tasks such as knowledge graph completion as compared to conventional methods. However, TransE neglects important multi-step path information in KGs and struggles with modeling complicated relations. For instance,  $(e_1, r_1, e_2)$  and  $(e_2, r_2, e_3)$  may reveal a new fact  $(e_1, r_1 \circ r_2, e_3)$ , where  $\circ$  is a certain function that composes the two relations.

As demonstrated in PTransE [Lin *et al.*, 2015], relation paths can significantly improve the performance of TransE. Following PTransE, we define the relation path as  $p = r_1 \circ r_2$  and  $P(h, t) = \{p | \forall e \in E, r_1, r_2 \in R, (h, r_1, e), (e, r_2, t) \in T, p = r_1 \circ r_2\}$ . In PTransE, if a relation path plays the same role as a relation, that is,  $p \in P(h, t)$  and  $(h, r, t) \in T$ , we define the path embedding as  $\mathbf{p} \simeq \mathbf{r}$ . The energy function is defined as:

$$E(p, r) = \|\mathbf{p} - \mathbf{r}\| = \|\mathbf{p} - (\mathbf{t} - \mathbf{h})\| = E(h, p, t). \quad (6)$$

As will be shown in our experiments, PTransE takes multi-step path information into consideration and can thus achieve better performances for entity alignment.

The score function of PTransE is defined as appending a path-relation regularization term to that of TransE:

$$\mathcal{K}_P = \mathcal{K}_T + \sum_{T \in \{T_1, T_2\}} \sum_{(h, r, t) \in T} \left[ \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t) L(p, r) \right], \quad (7)$$

where  $L(p, r)$  is a margin-based loss function with respect to the pair  $(p, r)$ . Following [Lin *et al.*, 2015],  $R(p|h, t)$  indicates the reliability of the path  $p$  given the entity pair  $(h, t)$ .  $Z = \sum_{p \in P(h, t)} R(p|h, t)$  is a normalization factor. We further define  $L(p, r)$  as a margin-based loss function:

$$L(p, r) = \sum_{(h, r', t) \in T^-} [\gamma + E(p, r) - E(p, r')]_+. \quad (8)$$

## 4.3 Joint Embeddings

The aforementioned knowledge embeddings are typically learned separately in different KGs. For entity alignment, we have to join these knowledge embeddings into a unified semantic space. We perform joint embeddings based on *alignment seeds* to accomplish this goal. We propose three models for joint embeddings, which are introduced in details as follows.

**Translation-based Model** Inspired by translation-based KRL methods such as TransE, it is straightforward to regard the *alignment* as a special relation between entities, and perform an alignment-specific translation operation between *aligned entities* to learn joint embeddings.

Formally, given two *aligned entities*  $e_1 \in E_1$  and  $e_2 \in E_2$ , we assume there is an alignment relation  $r^{(E_1 \rightarrow E_2)}$  so that  $\mathbf{e}_1 + \mathbf{r}^{(E_1 \rightarrow E_2)} \simeq \mathbf{e}_2$ . The energy function of joint embeddings is thus defined as:

$$E(e_1, e_2) = \|\mathbf{e}_1 + \mathbf{r}^{(E_1 \rightarrow E_2)} - \mathbf{e}_2\|. \quad (9)$$

**Linear Transformation Model** We can also learn a linear transformation between knowledge embeddings of various KGs. Given two *aligned entities*  $e_1 \in E_1$  and  $e_2 \in E_2$ , we define a transformation matrix  $\mathbf{M}^{(E_1 \rightarrow E_2)}$ , so that  $\mathbf{M}^{(E_1 \rightarrow E_2)}\mathbf{e}_1 \simeq \mathbf{e}_2$ . This idea and Translation-based (TB) Model have also been explored in [Chen *et al.*, 2016] for entity alignment. Hence, we define the energy function as:

$$E(e_1, e_2) = \|\mathbf{M}^{(E_1 \rightarrow E_2)}\mathbf{e}_1 - \mathbf{e}_2\|. \quad (10)$$

For both the Translation-based Model and the Linear Transformation (LT) Model, we can define the score function as the sum of energy functions over *alignment seeds*, which can be formalized as:

$$\mathcal{J}_{T/L} = \sum_{(e_1, e_2) \in \mathbb{L}} \alpha E(e_1, e_2), \quad (11)$$

where  $\alpha$  is a weighted factor.

**Parameter Sharing Model** Those two models mentioned above can be regarded as regularization to learning knowledge embeddings. Since *aligned entities* have identical meanings in KGs, it is also intuitive for us to make those *aligned entities* share the same embeddings. Formally, for each aligned entity pair  $(e_1, e_2)$ , we define:

$$\mathbf{e}_1 \equiv \mathbf{e}_2, (e_1, e_2) \in \mathbb{L}. \quad (12)$$

The Parameter Sharing (PS) Model is simple and effective to calibrate knowledge embeddings of  $KG_1$  and  $KG_2$  into the same semantic space. In this model, there are no regularization variables, so its score function  $\mathcal{J}_P = 0$ . The idea of parameter sharing has also been mentioned in [Goodfellow *et al.*, 2016] as an effective alternative to regularization for utilizing prior knowledge about variable dependency.

#### 4.4 Iterative Alignment

Based on knowledge embeddings and joint embeddings, we can perform entity alignment in the unified semantic space according to semantic distances between entities. The semantic distance is calculated in different ways with respect to various models of joint embeddings. For Translation-based Model and Linear Transformation Model, the distance is calculated using the energy functions defined in Eq. (9) and Eq. (10). For Parameter Sharing Model, the distance is calculated as  $E(e_1, e_2) = \|\mathbf{e}_1 - \mathbf{e}_2\|_{L1/L2}, \forall e_1 \in E_1, e_2 \in E_2$ .

In this way, for each non-aligned entity  $e_1$  in one KG, we find the nearest non-aligned entity  $\hat{e}_2$  from another KG:  $\hat{e}_2 = \arg \min_{e_2} (E(e_1, e_2))$ . We also define a distance threshold  $\theta$  as a hyper-parameter, and if  $E(e_1, \hat{e}_2) < \theta$ , we will be confident that  $\hat{e}_2$  is very likely to be the counterpart of  $e_1$ , otherwise we will not regard  $\hat{e}_2$  as the counterpart of  $e_1$ . We call these entities *newly aligned entities*.

It is intuitive that, *newly aligned entities* can help update joint embeddings and find more entities to align. Hence, we propose iterative entity alignment and design two strategies for iterative learning of joint embeddings and entity alignment.

**Hard Alignment** For Parameter Sharing Model, we can simply apply the rule of parameter sharing for those *newly aligned entities*. That is, we append each *newly aligned entity pair*  $(e_1, e_2)$  into the *alignment seeds*  $\mathbb{L}$ , and simply force  $\mathbf{e}_1$

and  $\mathbf{e}_2 \leftarrow \frac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2)$ . Afterwards, we update joint embeddings according to the updated seed set  $\mathbb{L}$ . For Hard Alignment, *newly aligned entity pairs* are directly added into  $\mathbb{L}$ , so the score function of Hard Alignment (HA) is  $\mathcal{I}_H = 0$ .

**Soft Alignment** Since there are inevitable errors in entity alignment, Hard Alignment may suffer from error propagation when introducing wrong alignments. For example, since both *George W. Bush* and *Bill Clinton* were presidents of *the United States*, they exhibit similar embeddings. Suppose *George W. Bush* in  $KG_1$  and *Bill Clinton* in  $KG_2$  are occasionally aligned by Hard Alignment, it may correspondingly force the embeddings of their birth places, *New Haven* in  $KG_1$  and *Hope* in  $KG_2$  much closer, leading to more errors.

To address this issue, we propose Soft Alignment (SA) by assigning a reliability score to each *newly aligned entities*. This method can be used accompanying with Translation-based Model, Linear Transformation Model and Parameter Sharing Model. Formally, we build a set  $\mathbb{M}$ , and iteratively add *newly aligned entity pairs* into this set. For each aligned entity pair  $(e_1, e_2)$ , we define a mapping  $R : (E_1, E_2) \rightarrow [0, 1]$  to calculate the reliability score:

$$R(e_1, e_2) = \sigma(k(\theta - E(e_1, e_2))), \quad (13)$$

where  $\sigma(\cdot)$  is the sigmoid function,  $k$  is a hyper-parameter satisfying  $k \in \mathbb{R}^+$ . It can be observed that, the reliability score corresponds to the semantic distance  $E(e_1, e_2)$ .

Based on reliability scores of *newly aligned entities*, we formalize the score function of Soft Alignment as:

$$\begin{aligned} \mathcal{I}_S &= \sum_{(e_1, e_2) \in \mathbb{M}} R(e_1, e_2)(\mathcal{H}_{(e_1, e_2)} + \mathcal{H}_{(e_2, e_1)}), \\ \mathcal{H}_{(e_1, e_2)} &= \sum_{(e_1, r, t)} U(e_2, r, t) + \sum_{(h, r, e_1)} U(h, r, e_2), \end{aligned} \quad (14)$$

where  $U(h, r, t)$  indicates the loss on this triple. For TransE and PTransE, we have different  $U$ : In TransE,  $U(h, r, t) = L(h, r, t)$ ; in PTransE,  $U(h, r, t) = L(h, r, t) + \frac{1}{Z} \sum_{p \in P(h, t)} R(p|h, t)L(p, r)$ .

#### 4.5 Optimization and Implementation Details

We use stochastic gradient descent (SGD) as our optimizer. In Linear Transformation Model, we initialize matrix following [Chen *et al.*, 2016]. The knowledge embeddings  $\mathbf{E} = \{\mathbf{e} | e \in E\}$  and  $\mathbf{R} = \{\mathbf{r} | r \in R\}$  are initialized by drawing from a normal distribution. In Soft Alignment, to achieve better quality, we limit the number of *newly aligned entities* in each alignment procedure to a threshold. We denote the epochs where we conduct alignment as  $B = \{b_i\}_{i=0}^{N_B}$ , and in the  $b_i$ -th epoch, we set the maximum number of alignments to  $c_i$ , and  $C = \{c_i\}_{i=0}^{N_C}$ . Note that  $N_B = N_C$ .

## 5 Experiment

In experiments, we mainly concentrate on entity alignment. Moreover, the learned knowledge representations could also help in mono-KG knowledge graph completion including entity prediction and relation prediction.

## 5.1 Datasets

In this paper, we build four datasets based on FB15K [Bordes *et al.*, 2013] originally extracted from Freebase [Bordes *et al.*, 2013], which contains 14,951 entities, 1,345 relations and 592,213 triples in total. The first three datasets, DFB-1, DFB-2, DFB-3, are for entity alignment, and the last DFB-4 is for knowledge graph completion.

**DFB-1, DFB-2 and DFB-3.** We build the three datasets in similar ways by randomly dividing FB15K triples into two subsets  $T_1$  and  $T_2$  of similar size, and making the overlap ratio  $O$  of the amount of shared triples between  $T_1$  and  $T_2$  to all triples satisfying pre-defined values. Entity set  $E$  and relation set  $R$  in these two KGs are the same. We know all relation alignments and *alignment seeds*  $\mathbb{L}$  selected from those most frequent entities. The alignments of other entities are used as the test set and validation set. The three datasets are built with different  $|\mathbb{L}|$  and overlap  $O$  between the two KGs, as shown in Table 1.

Table 1: Statistics of DFB-1, DFB-2 and DFB-3<sup>1</sup>.

Dataset	$ R $	$ E $	$ T_1 $	$ T_2 $	$ \mathbb{L} $	#Valid	$O$
DFB-1	1,345	14,951	444,159	444,160	5,000	1,000	0.5
DFB-2	1,345	14,951	444,159	444,160	500	1,000	0.5
DFB-3	1,345	14,951	325,717	325,717	500	1,000	0.1

**DFB-4.** We extract three triple sets from FB15k. The first two are for training and testing, and the third one is an auxiliary training set. We first extract a test set and divide the remaining triples into two subsets, a training set and an auxiliary training set, treated as triples of two KGs  $T_1$  and  $T_2$ . Entity set  $E$ , relation set  $R$  and *alignment seeds*  $\mathbb{L}$  are identical to DFB-1. The size of training set, test set and auxiliary training set are 399, 856/59, 071/399, 857 respectively.

## 5.2 Experiment Settings

For knowledge embeddings, the dissimilarity measure in TransE is implemented through L1-norm. Both TransE and PTransE are learned with the best parameters reported in their papers, and PTransE is “ADD, 2-STEP” version. For joint embeddings, the hyper-parameters in Translation-based Model and Linear Transformation Model are set to the best ones in [Chen *et al.*, 2016], except for embedding dimension  $n$  and learning rate  $\lambda$  which are set differently.

Our methods are trained with stochastic gradient descent (SGD). As for hyper-parameters, we select margin  $\gamma$  among  $\{0.5, 1.0, 1.5, 2.0\}$ . We set the dimensions of entity and relation embeddings to be the same  $n$ . We set a fixed learning rate  $\lambda = 0.001$  following [Bordes *et al.*, 2013; Lin *et al.*, 2015]. For Hard Alignment and Soft Alignment, we select  $\theta$  among  $\{0.5, 1.0, 2.0, 3.0, 4.0\}$ . For Soft Alignment, we select  $k$  among  $\{0.5, 1.0, 2\}$ . For a fair comparison, all models are trained under the same dimension  $n = 50$  and the same amount of epochs 3000. The optimal configurations of our models are:  $\gamma = 1.0, k = 1.0, B = \{1000, 1500, 2000, 2500\}, C =$

<sup>1</sup>Note that knowledge graphs in DFB-3 are more heterogenous than KGs in real world. Here DFB-3 is used only for testing the robustness of our models.

$\{5000, 6000, 7000, 8000\}, \theta = 1.0$  for Hard Alignment and  $\theta = 3.0$  for Soft Alignment. Note that, the threshold  $\theta$  is different for SA and HA because HA is more sensitive to alignment errors. These hyper-parameters are tuned for the first experiment, but the same configurations are also used in the second experiment.

In the following subsections, we name all our methods as “ITransE/IPTransE (HA/SA)” to denote different implementations including through TransE/PTransE and HA/SA. All our methods are implemented through PS Model because as will be shown later, PS Model outperforms LT and TB Models a lot.

## 5.3 Entities Alignment

Two measures are considered as evaluation metrics: (1) mean rank of correct entities or relations (Mean Rank); (2) proportion of correct answers ranked in top-10 and top-1 (Hits@10 and Hits@1). Thereof, we prefer higher Hits@10 and Hits@1 and lower Mean Rank that indicate better alignment.

For comparison, we select two models as our baselines: Translation-based Model and Linear Transformation Model based on TransE, which are adapted from “Var<sub>3</sub>” and “Var<sub>4</sub>” of MTransE in [Chen *et al.*, 2016], named “MTransE (TB)” and “MTransE (LT)”. We also select TransE/PTransE with Parameter Sharing Model (“TransE/PTransE + PS”) for comparison. Conventional graph-based models are not selected as baselines for their high computing complexity.

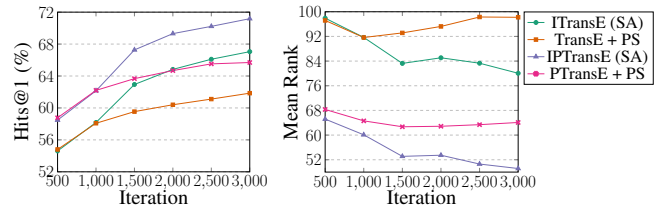


Figure 3: Hits@1 and Mean Rank of our methods through different iterations. (Hits@10 has similar trends to Hits@1.) Note that we conduct soft alignment every 500 iterations from the 1000-th iteration.

We show the evaluation results in Table 2. From the table, we can observe that: (1) Among various methods of joint embeddings, the Parameter Sharing Model outperforms both Linear Transformation Model and Translation-based Model. This verifies the fact that entities and their counterparts share the same intrinsic knowledge. (2) Among various methods of Iterative Alignment, Soft Alignment Model consistently outperforms Hard Alignment Model and other baselines. The reason is that Hard Alignment suffers from error propagation that we have analyzed before. (3) PTransE-based methods outperform the corresponding TransE-based methods. This indicates that better knowledge embeddings lead to more accurate alignment.

We further show the effectiveness of soft alignment strategy in each iteration in Fig. 3. From Fig. 3 we can observe that: (1) the performances of all methods increase with iterations with their increasing rates slowing down gradually. After the 3000-th iteration, the increasing rate is quite low, so we

Table 2: Results of entity alignment

Metric	DFB-1			DFB-2			DFB-3		
	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank	Hits@1	Hits@10	Mean Rank
MTransE (LT)	38.9	61.0	237.7	12.3	33.8	419.2	6.5	22.0	699.8
MTransE (TB)	13.6	35.1	547.7	13.9	35.4	675.7	4.5	16.1	1255.5
TransE + PS	61.9	79.2	105.2	41.1	67.0	154.9	12.2	34.6	431.9
ITransE (HA)	62.6	78.9	100.0	41.2	66.9	151.9	12.3	33.7	432.3
ITransE (SA)	<b>67.1</b>	<b>83.1</b>	<b>80.1</b>	<b>57.7</b>	<b>77.7</b>	<b>109.3</b>	<b>16.2</b>	<b>40.9</b>	<b>367.2</b>
PTransE + PS	65.8	83.4	62.9	46.3	72.1	96.8	15.8	40.2	346.9
IPTransE (HA)	66.1	83.3	59.1	46.2	72.6	94.2	15.1	39.7	337.6
IPTransE (SA)	<b>71.7</b>	<b>86.5</b>	<b>49.0</b>	<b>63.5</b>	<b>82.2</b>	<b>67.5</b>	<b>20.4</b>	<b>47.4</b>	<b>281.0</b>

Table 3: Results of knowledge graph completion (DFB-4)

Metric	Entity Prediction				Relation Prediction			
	Mean Rank		Hits@10		Mean Rank		Hits@1	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
MTransE (LT)	240.8	131.3	36.4	47.3	37.2	36.9	48.3	56.9
MTransE (TB)	851.3	759.7	9.4	10.8	293.7	293.4	27.4	27.7
TransE	246.1	131.6	42.5	54.3	55.9	55.6	44.2	50.7
TransE + Aux	232.8	121.5	43.3	54.9	50.1	49.8	44.4	50.9
ITransE (SA)	<b>209.2</b>	<b>101.0</b>	<b>44.2</b>	<b>55.1</b>	<b>19.8</b>	<b>19.6</b>	<b>54.2</b>	<b>60.7</b>
PTransE	213.0	97.2	50.9	72.1	2.33	1.96	67.4	86.9
PTransE + Aux	206.3	80.4	52.7	80.7	2.34	1.93	<b>68.8</b>	90.5
IPTransE (SA)	<b>197.5</b>	<b>70.6</b>	<b>53.0</b>	<b>80.8</b>	<b>2.03</b>	<b>1.62</b>	68.6	<b>90.8</b>

provide the result of the 3000-th iteration to balance the effectiveness and efficiency. And since the 1000-th iteration, SA methods achieve much better performances than their counterpart. (ITransE (SA) is even better than PTransE + PS at the 3000-th iteration.) (2) SA methods have a huge enhancement at the 1500-th iteration, which is probably due to the alignments we conducted at the 1000-th iteration. It can help entity pairs far from *alignment seeds* align properly, and thus enhances the overall performance. (3) For Mean Rank metric, non-iterative methods begin to increase at 1000-th to 1500-th iteration (indicating performance getting worse), while the trends of SA methods are descending. Since Mean Rank evaluates the overall performance of a model, the results confirm the robustness of our models.

#### 5.4 Knowledge Graph Completion

The above experiments have shown the capability of our methods for entity alignment task. We also want to show that entity alignment can also help learn better knowledge embeddings, which can be evaluated by knowledge graph completion [Bordes *et al.*, 2011; 2012; 2013]. Knowledge graph completion aims to complete a triple  $(h, r, t)$  when one of  $h, r, t$  is missing. Two measures are considered as our evaluation metrics: (1) mean rank of correct entities or relations (Mean Rank); (2) proportion of correct answers ranked in top-10 (Hits@10, for entities) or top-1 (Hits@1, for relations). We also follow the two evaluation settings named “raw” and “filter”. We conduct experiments on DFB-4 and divide the task into two subtasks: entity and relation prediction.

For comparison, we select several models as our baselines: (1) TransE/PTransE which can only utilize information from the training set of DFB-4, (2) “TransE/PTransE + Aux” which can utilize all triples from the training set and triples from the auxiliary set whose *head* and *tail* both are in *alignment seeds*, (3) “MTransE (LT)” and “MTransE (TB)”, as we introduced in the previous experiment. Since SA Models perform

better than HA Models and “TransE/PTransE + PS”, here we only select “ITransE/IPTransE (SA)” for comparison due to limited space.

The results of entity prediction and relation prediction are shown in Table 3. From the table we can observe that: (1) Parameter Sharing Model with Soft Alignment outperforms all baselines in almost all metrics only expect for comparable performance on “Raw” version of Hits@1. This result indicates that our methods can successfully utilize the information from auxiliary KG to improve knowledge embeddings. (2) Parameter Sharing Model also achieves much better Mean Rank as compared to the methods directly incorporating auxiliary KG (TransE/PTransE+Aux). This indicates that by iteratively adding *newly aligned entity pairs*, we can achieve more informative knowledge embeddings.

## 6 Conclusion and Future Work

This paper presents iterative entity alignment via joint knowledge embeddings, by encoding both entities and relations of various KGs into a unified semantic space. We propose a simple and effective Parameter Sharing Model and Iterative Alignment Model to learn joint embeddings and perform entity alignment simultaneously. We evaluate on entity alignment and knowledge graph completion, and experiment results show the effectiveness of our methods as compared with other baselines. The source code can be obtained from <https://github.com/thunlp/IEAKE>.

In future, we will explore the following research directions: (1) This paper only considers the internal structural information of KGs for entity alignment. In future, we will incorporate rich external information of KGs for entity alignment, and evaluate our models on real-world KG alignment. (2) There are many other effective KRL models, which can be easily adopted in our methods. In future work, we will explore the effectiveness of these KRL models in our methods for entity alignment.

## 7 Acknowledgement

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61661146007, 61572273), and Tsinghua University Initiative Scientific Research Program (20151080406).

## References

- [Ahn *et al.*, 2016] Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*, 2016.
- [Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*, pages 1247–1250, 2008.
- [Bordes *et al.*, 2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, pages 301–306, 2011.
- [Bordes *et al.*, 2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of AISTATS*, pages 127–135, 2012.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795, 2013.
- [Chen *et al.*, 2016] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multi-lingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*, 2016.
- [Dong *et al.*, 2014] Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of KDD*, pages 601–610, 2014.
- [Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.
- [Lacoste-Julien *et al.*, 2013] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. Sigma: Simple greedy matching for aligning large knowledge bases. In *Proceedings of SIGKDD*, pages 572–580. ACM, 2013.
- [Lehmann *et al.*, 2015] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.
- [Lin *et al.*, 2015] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of EMNLP*, pages 705–714, 2015.
- [Mahdisoltani *et al.*, 2014] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Proceedings of CIDR*, 2014.
- [Nguyen *et al.*, 2011] Thanh Nguyen, Viviane Moreira, Huong Nguyen, Hoa Nguyen, and Juliana Freire. Multilingual schema matching for wikipedia infoboxes. *Proceedings of VLDB*, 5(2):133–144, 2011.
- [Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*, pages 809–816, 2011.
- [Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of WWW*, pages 271–280, 2012.
- [Nickel *et al.*, 2016] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*, 2016.
- [Niu *et al.*, 2012] Xing Niu, Shu Rong, Haofen Wang, and Yong Yu. An effective rule miner for instance matching in a web of data. In *Proceedings of CIKM*, pages 1085–1094. ACM, 2012.
- [Pershina *et al.*, 2015] Maria Pershina, Mohamed Yakout, and Kaushik Chakrabarti. Holistic entity matching across knowledge graphs. In *Proceedings of IEEE International Conference on Big Data*, pages 1585–1590. IEEE, 2015.
- [Rinsler *et al.*, 2013] Daniel Rinsler, Dustin Lange, and Felix Naumann. Cross-lingual entity matching and infobox alignment in wikipedia. *Information Systems*, 38(6):887–907, 2013.
- [Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934, 2013.
- [Suchanek *et al.*, 2007] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW*, pages 697–706, 2007.
- [Vrandečić and Krötzsch, 2014] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
- [Wang *et al.*, 2013] Zhichun Wang, Juanzi Li, and Jie Tang. Boosting cross-lingual knowledge linking via concept annotation. In *IJCAI*, 2013.
- [Yin *et al.*, 2016] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering. In *Proceedings of IJCAI*, 2016.