Bootstrapping Entity Alignment with Knowledge Graph Embedding

Zequn Sun, Wei Hu[*]**, Qingheng Zhang** and **Yuzhong Qu**
State Key Laboratory for Novel Software Technology, Nanjing University, China
{zqsun, qhzhang}.nju@gmail.com, {whu, yzqu}@nju.edu.cn

Abstract

Embedding-based entity alignment represents different knowledge graphs (KGs) as low-dimensional embeddings and finds entity alignment by measuring the similarities between entity embeddings. Existing approaches have achieved promising results, however, they are still challenged by the lack of enough prior alignment as labeled training data. In this paper, we propose a bootstrapping approach to embedding-based entity alignment. It iteratively labels likely entity alignment as training data for learning alignment-oriented KG embeddings. Furthermore, it employs an alignment editing method to reduce error accumulation during iterations. Our experiments on real-world datasets showed that the proposed approach significantly outperformed the state-of-the-art embedding-based ones for entity alignment. The proposed alignment-oriented KG embedding, bootstrapping process and alignment editing method all contributed to the performance improvement.

1 Introduction

Recently, knowledge graphs (KGs) attract increasing attentions and are being widely used in AI-related applications, such as question answering, semantic search and knowledge reasoning. KGs contain a great amount of structured knowledge. A common representation of knowledge in KGs is in the form of triple, denoted by (h, r, t), indicating that there exists a relation r between its head entity h and tail entity t. To help capture the semantics hidden in KGs, a lot of research efforts have been devoted to KG embedding. Its key idea is to represent elements, such as entities and relations, in a KG as low-dimensional vectors (called embeddings) while preserving the inherent KG semantics. As an early work, TransE [Bordes *et al.*, 2013] interprets a relation as the translation from its head entity to its tail entity. It expects $\vec{v}(h) + \vec{v}(r) \approx \vec{v}(t)$ if (h, r, t) holds, where $\vec{v}(\cdot)$ denotes the embedding for a given element. TransE has demonstrated promising performance for KG completion. It is further improved by many works, such as TransH [Wang *et al.*, 2014], TransR [Lin *et al.*, 2015b] and PTransE [Lin *et al.*, 2015a].

Most existing KG embedding models focus on modeling a single KG. However, as applications that use KGs present themselves with more diversity, oftentimes a single KG can hardly satisfy various knowledge needs. An effective way to solve this problem is by integrating heterogeneous knowledge among different KGs via entity alignment. A few works study the problem of embedding different KGs towards entity alignment. MTransE [Chen *et al.*, 2017] performs cross-lingual entity alignment by spatially transforming monolingual KG embedding spaces. IPTransE [Zhu *et al.*, 2017] represents different KGs into a unified embedding space by configuring parameter sharing on existing alignment. JAPE [Sun *et al.*, 2017] refines KG embeddings for entity alignment by leveraging both relation and attribute embeddings.

Generally, the embedding-based approaches can provide several advantages compared with the conventional ones for entity alignment. They exploit inherent semantics, which is independent of the heterogeneity among KGs, such as different naming conventions, logical expressions and natural languages. On the contrary, the conventional approaches are not always effective because the symbolic nature of triples makes KGs hard to align. However, there still exist several challenges to embedding-based entity alignment. First, although the embedding models for a single KG have been extensively studied in the past few years, alignment-oriented KG embedding remains largely unexplored. Second, embedding-based entity alignment usually relies on existing entity alignment (called prior alignment in this paper) as training data. However, as argued in [Chen *et al.*, 2017], the accessible prior alignment usually accounts for a small proportion. The limited training data would prevent the embedding-based approaches from learning accurate embeddings for entity alignment. Thus, they often suffer from low precision.

To tackle the above challenges, we propose a bootstrapping approach for entity alignment in this paper. Bootstrapping is a widely-used semi-supervised learning technique [Yarowsky, 1995; Abney, 2004]. It iteratively trains a classifier by bootstrapping from both labeled and unlabeled data. Inspired by this idea, we iteratively label likely entity alignment as training data and leverage it for learning alignment-oriented embeddings. We summarize the main contributions of this paper as follows:

- We model entity alignment as a classification problem,

[*] Corresponding author

which seeks to maximize alignment likelihood over all labeled and unlabeled entities based on KG embeddings. (Section 3)

- For alignment-oriented KG embedding, we propose a limit-based objective function, which expects lower scores for positive triples while higher scores for negative triples. To sample less-distinguishable negative triples, we propose an ϵ-truncated uniform negative sampling method. We also swap aligned entities between the triples of different KGs to calibrate the embeddings in a unified space. (Section 4.1)
- To overcome the lack of enough training data, we propose a bootstrapping process, which updates alignment-oriented embeddings by labeling likely alignment and adding it into training data iteratively. It labels likely alignment based on a global optimal goal to guarantee the accuracy, and employs an alignment editing method to reduce error accumulation. (Section 4.2)
- We evaluated the proposed approach on three cross-lingual and two large-scale datasets. The experimental results showed that the proposed approach significantly outperformed three state-of-the-art approaches for entity alignment. Furthermore, the bootstrapping process can lead to 13%–18% improvement on precision. We also conducted analysis of key parameters and the accuracy of labeled likely alignment. The results showed that the proposed negative sampling, likely alignment labeling and editing methods all contributed to the performance improvement. (Section 5)

2 Related Work

2.1 KG Embedding

Learning embeddings for KGs has demonstrated its effectiveness in modeling the semantic information of KGs. TransE [Bordes *et al.*, 2013] interprets a relation as the translation from its head entity to its tail entity. This translation-based KG embedding model has shown its feasibility for KG completion and later is improved by many following studies. For example, TransH [Wang *et al.*, 2014] and TransR [Lin *et al.*, 2015b] extend TransE on modeling multi-mapping relations (e.g., one-to-many relations). Also, there exist quite a number of non-translational models for KG embedding, such as [Nickel *et al.*, 2011; Socher *et al.*, 2013; Yang *et al.*, 2017; Dettmers *et al.*, 2018].

Moreover, extra knowledge in KGs is leveraged by a few works for improving embedding. PTransE [Lin *et al.*, 2015a] incorporates reverse triples and relation paths. KR-EAR [Lin *et al.*, 2016] introduces categorical attributes (e.g., *gender*). Besides, type information, local structure of entities and global patterns are explored in [Krompaß*et al.*, 2015; Ristoski and Paulheim, 2016; Cochez *et al.*, 2017].

2.2 Entity Alignment

Automated entity alignment approaches leverage various features of KGs, such as the semantics of OWL properties [Hu *et al.*, 2011], compatible neighbors and attribute values of entities [Suchanek *et al.*, 2012] and structural information of relations [Lacoste-Julien *et al.*, 2013]. To overcome the heterogeneity between different KGs, a few approaches also make use of external lexicons, machine translation, Wikipedia links [Suchanek *et al.*, 2012; Wang *et al.*, 2013], etc. Unlike them, our approach is based on KG embeddings and independent of extra resources.

Recently, several embedding-based approaches for entity alignment were proposed. MTransE [Chen *et al.*, 2017] uses TransE to represent different KGs as independent embeddings, and learns transformation between KGs via five alignment models. IPTransE [Zhu *et al.*, 2017] employs PTransE [Lin *et al.*, 2015a] to embed a single KG and integrates three modules (translation-based, linear transformation and parameter sharing) for jointly embedding different KGs. It also uses newly-aligned entities to update embeddings iteratively. However, newly-aligned entities are found based on a local optimal distance measure, which relies heavily on the alignment precision. As it is difficult to guarantee the precision when priori alignment is limited, errors would accumulate during iterations. Thus, IPTransE expects relation alignment and a large portion of entity alignment to be known ahead of time to guarantee the accuracy of embeddings. JAPE [Sun *et al.*, 2017] learns embeddings for entities and relations of different KGs in a unified embedding space. It also embeds attributes and leverages attribute correlations to refine entity embeddings. However, when the attributes are heterogeneous and correlations are vague between KGs, the effectiveness of attribute embeddings would greatly reduce.

3 Problem Formulation

Let $\mathbf{X}$ be the entity set of KG_1 and $\mathbf{Y}$ be the entity set of KG_2. Entity alignment aims to find $\mathbf{A} = \{(x, y) \in \mathbf{X}\times\mathbf{Y} | x \sim_{\mathcal{R}} y\}$ for which an equivalence relation $\sim_{\mathcal{R}}$ holds between x and y. In some cases, a subset $\mathbf{A}'$ of $\mathbf{A}$ is already known, which can be used as prior alignment (training data). Let $\mathbf{X}' \subseteq \mathbf{X}$ and $\mathbf{Y}' \subseteq \mathbf{Y}$ denote the unaligned entities that we aim to align.

We model entity alignment as a classification problem of using entities in $\mathbf{Y}$ to label entities in $\mathbf{X}$. By convention, we consider the so-called one-to-one entity alignment: an entity can be associated with at most one label, and a label can be assigned to at most one entity [Lacoste-Julien *et al.*, 2013; Zhang *et al.*, 2015]. This one-to-one constraint sets our problem apart from the common classification problem. Let Θ denote the embeddings of KG_1 and KG_2, we define the probability of assigning label $y \in \mathbf{Y}$ to entity $x \in \mathbf{X}$ w.r.t. Θ, denoted by $\pi(y|x;\Theta)$, as follows:

$$\pi(y|x;\Theta) = \sigma\big(\text{sim}(\vec{v}(x), \vec{v}(y))\big), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\text{sim}(\cdot,\cdot)$ is a similarity measure. In this paper, we simply use cosine similarity measure, i.e., $\text{sim}(\vec{v}(x), \vec{v}(y)) = \frac{\vec{v}(x)\cdot\vec{v}(y)}{||\vec{v}(x)||_2||\vec{v}(y)||_2}$. The maximum-likelihood criterion guides us to choose the optimal $\hat{\Theta}$ that achieves the highest alignment likelihood:

$$\begin{aligned}\hat{\Theta} &= \arg\max_{\Theta} \sum_{x\in\mathbf{X}} \log \pi(\mathbb{L}_x|x;\Theta) \\ &= \arg\max_{\Theta} \sum_{x\in\mathbf{X}} \sum_{y\in\mathbf{Y}} \mathbf{1}_{[y=\mathbb{L}_x]} \log \pi(y|x;\Theta),\end{aligned} \tag{2}$$

where $\mathbb{L}_x$ denotes the true label (aligned counterpart) of entity x, and $\mathbf{1}_{[\cdot]}$ is an indicator function that denotes the truth value (1 or 0) of the given proposition.

Solving Eq. (2) to parameterize Θ is not easy. Let us consider a situation in which we force aligned entities to share the same embedding, i.e., $\vec{v}(\mathbb{L}_x) = \vec{v}(x)$, in the initialization process. In this way, the initialized embeddings achieve the highest alignment likelihood over labeled entities without any training. But this does not work for entity alignment, as it fails to preserve any information of unlabeled entities. For embedding-based entity alignment, both the inherent semantics of each KG and the common semantics shared by different KGs are useful information [Sun *et al.*, 2017]. Thus, the embeddings should not only capture the alignment likelihood, but also model the semantics of KGs. Furthermore, due to the inadequacy of prior alignment, it is insufficient to only observe the likelihood of labeled entities. A better solution may be labeling unlabeled entities to expand the training data and observing the alignment likelihood over all the entities.

4 Methodology

4.1 Alignment-Oriented KG Embedding

Our alignment-oriented KG embedding model aims to encode different KGs into a unified embedding space, such that the alignment likelihood between entities can be directly measured via their embeddings. It captures the semantics hidden in KGs and is free of symbolic heterogeneity.

In a single KG, the diversified relations between entities characterize its semantics. The translation-based models have shown their success in modeling KG semantics. They define score function $f(\tau) = \|\vec{v}(h) + \vec{v}(r) - \vec{v}(t)\|_2^2$ to measure the plausibility of triple $\tau = (h, r, t)$. They optimize the margin-based ranking loss to make the scores of positive triples lower than those of negative ones. However, as studied in [Zhou *et al.*, 2017], this loss function cannot ensure that the scores of positive triples are absolutely low to fulfill the translation. For entity alignment, absolutely low scores of positive triples help reduce the drift of embeddings in the unified space and better capture the common semantics of different KGs. Therefore, we propose a new objective function, denoted by $\mathcal{O}_e$, based on the limit-based loss [Zhou *et al.*, 2017]:

$$\mathcal{O}_e = \sum_{\tau \in \mathbb{T}^+} [f(\tau) - \gamma_1]_+ + \mu_1 \sum_{\tau' \in \mathbb{T}^-} [\gamma_2 - f(\tau')]_+, \tag{3}$$

where $[\cdot]_+ = \max(\cdot, 0)$. $\gamma_1, \gamma_2 > 0$ are two hyper-parameters and $\mu_1 > 0$ is a balance hyper-parameter. $\mathbb{T}^+$ and $\mathbb{T}^-$ denote the sets of positive and negative triples, respectively.

The proposed objective function has two desirable properties. First, positive triples are expected to have low scores while negative triples are expected to have high scores, i.e., $f(\tau) \leq \gamma_1$ and $f(\tau') \geq \gamma_2$. Thus, we can directly control the absolute scores of positive and negative triples as needed. In practice, we set $\gamma_2 > \gamma_1$ and γ_1 is a small positive value. Second, we have $f(\tau') - f(\tau) \geq \gamma_2 - \gamma_1$, which indicates that the proposed objective function still preserves the characteristic of the margin-based ranking loss.

ϵ-Truncated Uniform Negative Sampling

A widely-used sampling method to generate negative triples $\mathbb{T}^-$ is by the uniform negative sampling [Bordes *et al.*, 2013; Lin *et al.*, 2015a; Chen *et al.*, 2017; Zhu *et al.*, 2017; Sun *et al.*, 2017]. Given a triple $(h, r, t) \in \mathbb{T}^+$, it replaces either h or t with an arbitrary entity. However, the replacers sampled in this way may be easily distinguished from their originals. For example, if we sample a negative triple (*Tim Berners-Lee, capital of, USA*) for (*Washington DC, capital of, USA*), we can find that replacer *Tim Berners-Lee* and its original *Washington DC* are totally orthogonal, so that it would contribute little to embedding learning. Instead, *New York* may be a good replacer for *Washington DC*. Then, our alignment-oriented KG embedding model is expected to be capable of distinguishing the two similar (but one positive and one negative) triples.

Given entity x to be replaced, unlike the uniform negative sampling method that samples its replacer from all entities, we limit the sampling scope to a group of candidates. Specifically, we choose its s-nearest neighbors in the embedding space as candidates, where $s = \lceil (1 - \epsilon)N \rceil$, $\epsilon \in [0, 1)$ is a proportion, N is the number of entities in the KG, and $\lceil \cdot \rceil$ is the ceiling function. In this way, other entities having low similarities with x are truncated and would not be sampled; while less-distinguishable replacers with similar features (e.g., types, relations) are kept. Here, we use the cosine similarity between embeddings to search for x's neighbors.

Parameter Swapping

To leverage prior alignment $\mathbf{A}'$ for bridging different KGs, we swap aligned entities in their triples to calibrate the embeddings of KG$_1$ and KG$_2$ in the unified embedding space. Given an aligned entity pair $(x, y) \in \mathbf{A}'$, we generate the following supervised triples:

$$\begin{aligned}\mathbb{T}^s_{(x,y)} = &\{(y, r, t) | (x, r, t) \in \mathbb{T}^+_1\} \cup \{(h, r, y) | (h, r, x) \in \mathbb{T}^+_1\} \\ &\cup \{(x, r, t) | (y, r, t) \in \mathbb{T}^+_2\} \cup \{(h, r, x) | (h, r, y) \in \mathbb{T}^+_2\},\end{aligned} \tag{4}$$

where $\mathbb{T}^+_1$ and $\mathbb{T}^+_2$ denote the positive triple sets of KG$_1$ and KG$_2$, respectively. In total, we have $\mathbb{T}^+ = \mathbb{T}^+_1 \cup \mathbb{T}^+_2 \cup \mathbb{T}^s$ in Eq. (3), where $\mathbb{T}^s = \bigcup_{(x,y) \in \mathbf{A}'} \mathbb{T}^s_{(x,y)}$. Then, we sample negative triples $\mathbb{T}^-$ for $\mathbb{T}^+$.

4.2 Bootstrapping Alignment

The embedding-based entity alignment often suffers from inadequate prior alignment. To cope with this, we leverage the bootstrapping idea. Specifically, we iteratively label likely alignment as training data and use it to further improve entity embeddings and alignment.

Likely Alignment Labeling and Editing

Conventional bootstrapping methods usually choose the most confident label to label instances. At the t-th iteration, they set $\hat{y} = \arg\max_y \pi(y|x; \Theta^{(t)})$ such that $\pi(y|x; \Theta^{(t)}) > \gamma_3$, and label x as $\hat{y}$. γ_3 is a threshold in $[0, 1)$. However, since the labeled training data is limited, these methods usually cannot provide predictions with high confidence. Thus, the labeling process may be error-prone.

Towards our objective of maximizing alignment likelihood and obeying the one-to-one alignment constraint, we choose to label alignment at the t-th iteration by solving the following optimization problem:

$$\max \sum_{x \in \mathbf{X}'} \sum_{y \in \mathbf{Y}'_x} \pi(y|x;\Theta^{(t)}) \cdot \psi^{(t)}(x,y),$$
$$\text{s.t.} \sum_{x' \in \mathbf{X}'} \psi^{(t)}(x',y) \leq 1, \sum_{y' \in \mathbf{Y}'_x} \psi^{(t)}(x,y') \leq 1, \forall x,y \tag{5}$$

where $\Theta^{(t)}$ denotes the entity embeddings at the t-th iteration. $\mathbf{Y}'_x = \{y|y \in \mathbf{Y}' \text{ and } \pi(y|x;\Theta^{(t)}) > \gamma_3\}$ denotes the candidate labels of x. $\psi^{(t)}(\cdot)$ is an indicator function that we seek to solve. $\psi^{(t)}(x,y) = 1$ if x is labeled as y at the t-th iteration, and 0 otherwise. The two constraints guarantee the one-to-one labeling. After solving Eq. (5), we get the newly-labeled alignment at the t-th iteration $\mathbf{A}^{(t)} = \{(x,y) \in \mathbf{X}' \times \mathbf{Y}' | \psi^{(t)}(x,y) = 1\}$. We use the newly-labeled alignment in an incremental manner and leverage it to guide the subsequent training.

Although the alignment is expected to improve over time, the labeling process may still create erroneous labels, which would misguide the subsequent training. Furthermore, labeling conflicts are inevitable when we accumulate the newly-labeled alignment of different iterations. In order to improve the labeling quality and satisfy the one-to-one alignment constraint, in our bootstrapping process, an entity once labeled can be relabeled or become unlabeled in the subsequent iterations. We employ a simple but effective editing technique to achieve this manner.

During bootstrapping, the newly-labeled alignment would be checked to see whether it leads to conflicts. For example, let us consider the case where one entity has conflict labels in different iterations. Assume that we have two candidate labels y and y' for entity x, we would like to choose the one that provides more alignment likelihood to label x. Formally, we calculate the following likelihood difference:

$$\Delta^{(t)}_{(x,y,y')} = \pi(y|x;\Theta^{(t)}) - \pi(y'|x;\Theta^{(t)}). \tag{6}$$

$\Delta^{(t)}_{(x,y,y')} > 0$ indicates labeling x as y gives more alignment likelihood than y'. Hence, we choose y to label x. The conflict where one label is assigned to multiple entities can be solved in the same way.

Learning from Holistic Perspective

To obtain a holistic observation of both labeled and unlabeled entities, we define a probability distribution ϕ_x to describe all the labeling possibilities of x [Abney, 2004]. Specifically, if x is labeled as $\hat{y}$, the labeling distribution ϕ_x has all of its mass concentrated on $\hat{y}$, while if x is unlabeled, ϕ_x is the uniform distribution. For entity x in the prior alignment, we have $\hat{y} = \mathbb{L}_x$. Formally, we have:

$$\phi_x(y) = \begin{cases} \mathbf{1}_{[y=\hat{y}]} & \text{if } x \text{ is labeled as } \hat{y} \\ \frac{1}{|\mathbf{Y}'|} & \text{if } x \text{ is unlabeled} \end{cases}. \tag{7}$$

Given this probability distribution, we minimize the following negative log-likelihood function to obtain the optimal embeddings Θ with the highest alignment likelihood:

$$\mathcal{O}_a = -\sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{Y}} \phi_x(y) \log \pi(y|x;\Theta). \tag{8}$$

Datasets		# Ent.	# Rel.	# Attr.	# Rel tr.	# Attr tr.
DBP-WD	DBpedia	100,000	330	351	463,294	381,166
	Wikidata	100,000	220	729	448,774	789,815
DBP-YG	DBpedia	100,000	302	334	428,952	451,646
	YAGO3	100,000	31	23	502,563	118,376

Table 1: Statistical data of DWY100K

Recall that embeddings Θ should not only capture the alignment likelihood, but also model the semantics of KGs. Thus, we define the joint objective function as follows:

$$\mathcal{O} = \mathcal{O}_e + \mu_2 \cdot \mathcal{O}_a, \tag{9}$$

where μ_2 is a hyper-parameter for balance.

4.3 Implementation Details

We initialize the embeddings of KGs based on the normal distribution, and use the gradient descent optimization algorithm Ada-Grad [Duchi *et al.*, 2011] to optimize Eq. (9). The length of all embeddings is restrained to 1 to avoid trivially optimizing the objective by increasing the norm of embeddings.

Solving Eq. (5) can be transformed to the max-weighted matching problem on bipartite graphs. We first pick (x,y) pairs satisfying their likelihood $\pi(y|x;\Theta^{(t)}) > \gamma_3$, and then build a bipartite graph whose nodes denote entities and edges have weights representing the alignment likelihood between nodes. Therefore, labeling likely alignment with the maximum likelihood can be solved by finding disjoint edges with the maximum total weight in the bipartite graph.

For the complexity of our approach, the number of parameters is DM, where D denotes the dimension of embeddings and M denotes the number of all the entities and relations in two KGs. For ϵ-truncated uniform negative sampling, searching for the s-nearest neighbors for one entity averagely takes linear time using the quick select algorithm. The time complexity of solving Eq. (5) can be reduced to linear time using the heuristic algorithm in [Hendrickson and Leland, 1995].

5 Experiments

We used Tensorflow to develop our approach, called BootEA. Our experiments were conducted on a personal workstation with an Intel Xeon E3 3.3 GHz CPU, a NVIDIA GeForce GTX 1080 Ti GPU and 128 GB memory. Our source code, datasets and experimental results are available online[1].

5.1 Datasets

In order to evaluate BootEA on various entity alignment scenarios, we used the following datasets in our experiments:

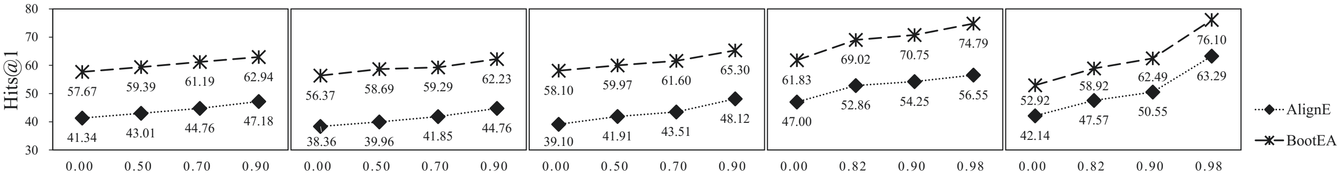
- DBP15K [Sun *et al.*, 2017] contains three cross-lingual datasets built from the multilingual versions of DBpedia: DBP$_{\text{ZH-EN}}$ (Chinese to English), DBP$_{\text{JA-EN}}$ (Japanese to English) and DBP$_{\text{FR-EN}}$ (French to English). Each dataset contains 15 thousand reference entity alignment.
- DWY100K contains two large-scale datasets extracted from DBpedia, Wikidata and YAGO3, denoted by DBP-WD and DBP-YG. Each dataset has 100 thousand reference entity alignment. The extraction method followed

[1] https://github.com/nju-websoft/BootEA

Approaches	$\text{DBP}_{\text{ZH-EN}}$			$\text{DBP}_{\text{JA-EN}}$			$\text{DBP}_{\text{FR-EN}}$			DBP-WD			DBP-YG		
	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
IPTransE	40.59	73.47	0.516	36.69	69.26	0.474	33.30	68.54	0.451	34.85	63.84	0.447	29.74	55.76	0.386
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
AlignE	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
BootEA	**62.94**	**84.75**	**0.703**	**62.23**	**85.39**	**0.701**	**65.30**	**87.44**	**0.731**	**74.79**	**89.84**	**0.801**	**76.10**	**89.44**	**0.808**

"*" denotes the unreported metrics in their papers. We reproduced the results using their source code.

Table 2: Result comparison on entity alignment



(A) $\text{DBP}_{\text{ZH-EN}}$ (B) $\text{DBP}_{\text{JA-EN}}$ (C) $\text{DBP}_{\text{FR-EN}}$ (D) DBP-WD (E) DBP-YG

Figure 1: Hits@1 on entity alignment w.r.t. ϵ-truncated uniform negative sampling

that for DBP15K. Taking DBP-WD for example, we first randomly extracted 100 thousand reference entity alignment from the English version of DBpedia to Wikidata. Each entity must be involved in at least one relation triple. We then extracted all the triples that only involve the entities in the alignment. The statistical data of DWY100K is listed in Table 1. Note that DBP-YG has imbalanced relation numbers, which would bring more challenges to embedding-based entity alignment.

5.2 Experiment Settings

For comparison, we chose three state-of-the-art embedding-based approaches to entity alignment, which have been discussed in Section 2. The following is their implementation details in our experiments.

- MTransE [Chen *et al.*, 2017] integrates five variants in its alignment model, where the fourth performs best according to its authors. We chose it to represent MTransE.
- IPTransE [Zhu *et al.*, 2017] is an iterative approach. We selected its best variant with parameter sharing and iterative alignment.
- JAPE [Sun *et al.*, 2017] combines relation and attribute embeddings for entity alignment. We used its full model.

For BootEA, we used the configuration below: $\gamma_1 = 0.01$, $\gamma_2 = 2.0$, $\gamma_3 = 0.7$, $\mu_1 = 0.2$ and $\mu_2 = 0.1$. Also, $\epsilon = 0.9$ for DBP15K and $\epsilon = 0.98$ for DWY100K. 10 negative triples were sampled for each positive triple. The learning rate was set to 0.01 and the training spent 500 epochs. One iteration of bootstrapping was conducted when training 10 epochs of embeddings. Thus, the iteration number of bootstrapping is 50. In our negative sampling, we generate the s-nearest candidates for each entity once every 5 epochs. Parameter settings for the comparative approaches followed their papers. For all the approaches, the embedding dimensions were set to 75 for a fair comparison.

By convention, we chose Hits@k and mean reciprocal rank (MRR) as our metrics. Hits@k measures the percentage of correct alignment ranked at top k. MRR is the average of the reciprocal ranks of results. Note that Hits@1 equates to precision. Higher Hits@k and MRR indicate better performance.

Following JAPE [Sun *et al.*, 2017], we used 30% of the reference entity alignment as prior alignment and left the remaining as testing data. For an ablation study, we separated a variant, called AlignE, from BootEA. AlignE is the implementation of our alignment-oriented KG embedding model with ϵ-truncated uniform negative sampling and parameter swapping. It optimizes Eq. (3) without bootstrapping. We ran AlignE and BootEA five times and reported the average.

5.3 Entity Alignment

Table 2 lists the results of entity alignment on DBP15K and DWY100K. We observed that AlignE clearly outperformed MTransE, IPTransE and JAPE, due to its alignment-oriented embedding. For MTransE, information loss happened when it learned the transformation between different embedding spaces. IPTransE and JAPE got better results than MTranE, because they leveraged relation paths and entity attributes, respectively. On large-scale datasets DBP-WD and DBP-YG, AlignE performed even better than on DBP15K, due to large-scale datasets provide richer semantics.

BootEA considerably improved the results of AlignE after employing bootstrapping, especially on Hits@1. For example, Hits@1 on DBP-WD was raised from 56.55% to 74.79%, which demonstrated the real precision of our approach. The good performance of bootstrapping is due to its capability of accurately labeling likely alignment as training data. We believe that our bootstrapping process can be a general enhancement for the embedding-based alignment approaches.

5.4 Analysis

Effectiveness of ϵ-Truncated Uniform Negative Sampling

To evaluate our negative sampling method, we tested ϵ among $\{0, 0.5, 0.7, 0.9\}$ for DBP15K and $\{0, 0.82, 0.9, 0.98\}$ for DWY100K. When $\epsilon = 0$, it is the uniform negative sampling.

From the results shown in Figure 1, we found that AlignE with the uniform negative sampling still obtained superior results compared to MTransE, IPTransE and JAPE (see the results in Table 2). Furthermore, the ϵ-truncated uniform negative sampling method brought large improvement for AlignE, which demonstrated that this sampling method worked well

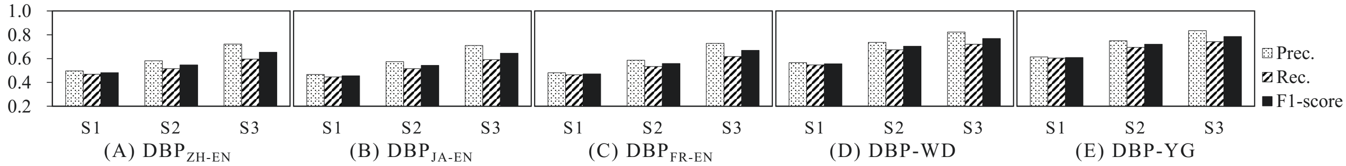


Figure 2: Precision, recall and F1-score of likely alignment w.r.t. different labeling methods

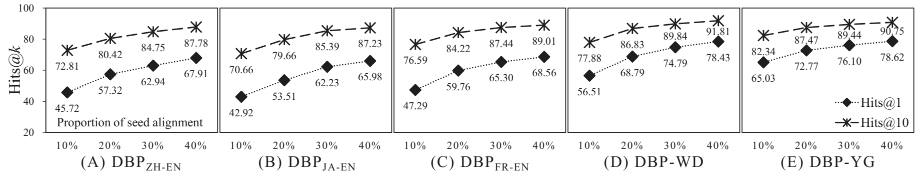


Figure 3: Hits@k on entity alignment w.r.t. proportion of prior alignment

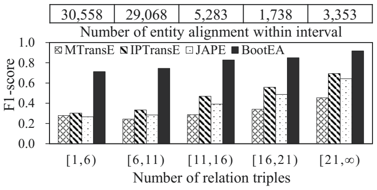


Figure 4: F1-score w.r.t. triple numbers

for entity alignment. Particularly, it greatly improved both AlignE and BootEA on DBP-YG. We believe that this is due to the small number of relations (31) in YAGO3. Having such a small number of relations but vast relation triples $(502, 563)$ indicates that YAGO3 contains plenty of multi-mapping relations (73.8%), which are hard for the translation-based models [Wang *et al.*, 2014; Lin *et al.*, 2015b]. Our method sampled less-distinguishable negative triples for AlignE, which helped model multi-mapping relations. Also, we found that different ϵ values made different contributions. Generally, a relatively large value is recommended, e.g., 0.9 for DBP15K and 0.98 for DWY100K. For large-scale KGs, it is also better to use a large ϵ due to the large cardinal number of entities.

Accuracy of Likely Alignment

To evaluate the accuracy of final labeled likely alignment, we compared three different labeling methods. S1 denotes the method used by conventional bootstrapping approaches and IPTransE. It firstly sets $\hat{y} = \arg\max_y \pi(y|x;\Theta^{(t)})$ s.t. $\pi(y|x;\Theta^{(t)}) > \gamma_3$, and then labels x as $\hat{y}$. S2 labels alignment by solving Eq. (5). It is not a cumulative process and does not use alignment editing. S3 accumulates the results of S2 with editing. It is the labeling method used in BootEA.

The comparison results are illustrated in Figure 2. We can see that our labeling method S3 performed best. The conventional labeling method S1 failed to achieve promising results. For example, its F1-score on DBP$_{\text{JA-EN}}$ only reached 0.4558. This is due to the poor robustness of S1. As it finds newly-aligned entities based on a local optimal measure, it would introduce many noisy labels when KG embeddings are not well trained or have low precision (Hits@1) for entity alignment. By labeling with a global optimal goal and imposing the one-to-one constraint, S2 improved largely, which was further enhanced with our editing method. These results verified that our method can guarantee the safeness for the use of unlabeled data. For the running time, the labeling per iteration averagely spent 9.273s on DBP15K and 48.852s on DWY100K. The editing took 0.035s and 0.231s, respectively.

Sensitivity to Proportion of Prior Alignment

To analyze whether BootEA is sensitive to the proportion of initial prior alignment, we tested the proportion from 10% to 40% with step 10%. Figure 3 depicts the changes of Hits@k with different values. As expected, the results on all the five datasets became better with the increase of the proportion, because more prior alignment can provide more information to align two KGs. When only using 10% of the gold standard as the prior alignment, BootEA still achieved satisfactory results. For example, Hits@10 on the five datasets are 72.81%, 70.66%, 76.59%, 77.88% and 82.34%, respectively. The results showed the robustness of BootEA.

F1-score w.r.t. Distribution of Relation Triple Numbers

For an entity, the number of its relation triples represents the richness of its content. Intuitively, it should be easier to align entities with rich content. In this analysis, we divided entity links in testing data into several intervals based on the number of their relation triples. For a testing entity link, we took the average number of relation triples of the two involved entities as its triple number. The performance was assessed by F1-score within a certain interval. Due to lack of space, we only reported the results on DBP-WD in Figure 4.

We found that BootEA outperformed MTransE, IPTransE and JAPE on all the intervals, which again confirmed the effectiveness of BootEA. In line with our expectations, the F1-score of all the approaches increased with the growth of relation triple numbers. However, the performance gap between BootEA and other approaches on sparse entities is larger than dense ones. For example, in interval $[1, 6)$, the difference between BootEA and IPTransE is 0.4103, while in interval $[21, \infty)$, the gap is 0.2246. This analysis demonstrated that BootEA can achieve promising results on sparse data, indicating its practical use for real KGs.

6 Conclusion and Future Work

The main contributions of this paper are threefold: (1) we introduced a KG embedding model to learn alignment-oriented embeddings across different KGs. It employs an ϵ-truncated uniform negative sampling method to improve alignment performance; (2) we conducted entity alignment in a bootstrapping process. It labels likely alignment as training data and edits alignment during iterations; and (3) our experiment results showed that the proposed approach significantly outperformed three state-of-the-art embedding-based ones, on three cross-lingual datasets and two new large-scale datasets constructed from DBpedia, Wikidata and YAGO3. For future work, we plan to study the cross-lingual word embeddings for attribute values. Also, we want to leverage the recurrent neural networks for modeling the complex semantics of KGs.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2018YFB1004300). Part of this work was done during the second author's visit to University of Toronto.

References

[Abney, 2004] Steven P. Abney. Understanding the yarowsky algorithm. *Computational Linguistics*, 30(3):365–395, 2004.

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

[Chen *et al.*, 2017] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*, pages 1511–1517, 2017.

[Cochez *et al.*, 2017] Michael Cochez, Petar Ristoski, Simone Paolo Ponzetto, and Heiko Paulheim. Global RDF vector space embeddings. In *Proceedings of ISWC*, pages 190–207, 2017.

[Dettmers *et al.*, 2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D knowledge graph embeddings. In *Proceedings of AAAI*, 2018.

[Duchi *et al.*, 2011] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[Hendrickson and Leland, 1995] Bruce Hendrickson and Robert Leland. A multilevel algorithm for partitioning graphs. In *Proceedings of Supercomputing*, 1995.

[Hu *et al.*, 2011] Wei Hu, Jianfeng Chen, and Yuzhong Qu. A self-training approach for resolving object coreference on the semantic web. In *Proceedings of WWW*, pages 87–96, 2011.

[Krompaß*et al.*, 2015] Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *Proceedings of ISWC*, pages 640–655, 2015.

[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 KDD*, pages 572–580, 2013.

[Lin *et al.*, 2015a] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of ACL*, pages 705–714, 2015.

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

[Lin *et al.*, 2016] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. In *Proceedings of IJCAI*, pages 2866–2872, 2016.

[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.

[Ristoski and Paulheim, 2016] Petar Ristoski and Heiko Paulheim. RDF2Vec: RDF graph embeddings for data mining. In *Proceedings of ISWC*, pages 498–514, 2016.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934, 2013.

[Suchanek *et al.*, 2012] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. PARIS: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*, 5(3):157–168, 2012.

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

[Wang *et al.*, 2013] Zhichun Wang, Juanzi Li, and Jie Tang. Boosting cross-lingual knowledge linking via concept annotation. In *Proceedings of IJCAI*, pages 2733–2739, 2013.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, pages 1112–1119, 2014.

[Yang *et al.*, 2017] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Proceedings of NIPS*, pages 2316–2325, 2017.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*, pages 189–196, 1995.

[Zhang *et al.*, 2015] Duo Zhang, Benjamin I. P. Rubinstein, and Jim Gemmell. Principled graph matching algorithms for integrating multiple data sources. *IEEE Trans. Knowl. Data Eng.*, 27(10):2784–2796, 2015.

[Zhou *et al.*, 2017] Xiaofei Zhou, Qiannan Zhu, Ping Liu, and Li Guo. Learning knowledge embeddings by combining limit-based scoring loss. In *Proceedings of CIKM*, pages 1009–1018, 2017.

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