Dynamic Alignment of Entities, Relations and Attributes via Heterogeneous Knowledge Base Embedding

Abstract

We study the general problem of knowledge base (KB) alignment, which aims to align entities, relations and attributes between different KBs. While existing approaches exploit various features of KBs and external resources to deal with this problem, they still face challenges in tackling multifarious semantic heterogeneities and cross-linguality. Moreover, most of them separate the entity alignment process from relation and attribute alignment process. This paper proposes a new embedding-based approach to KB alignment. Specifically, we introduce a heterogeneous KB embedding (HKBE) module to encode entities, relations and attributes in different KBs into a unified vector space while preserving their inherent proximity. We also design a dynamic alignment (DA) module that leverages potentially-aligned entities as supervision to improve the learning accuracy. Our experiments on real-world KBs show that our approach significantly outperforms several state-of-the-art embedding-based methods for entity alignment. It also achieves promising results on relation alignment and attribute alignment.

Introduction

Knowledge bases (KBs) such as DBpedia and Wikidata store rich facts about the real world. Each fact is expressed in a structured form, which is usually a triple of (*entity, property, value*). There are two kinds of properties, namely relations and attributes. Relations have entities as values, while attributes have literals as values. Structured KBs have been widely used in AI-related applications, e.g., semantic search, question answering and knowledge reasoning. As these applications become more popular and diverse, they put forward higher demands on KBs, in terms of coverage, richness and multilingualism. Oftentimes a single KB cannot meet all these demands. This difficulty calls for the integration of multiple KBs.

However, as KBs are usually developed by independent parties using different domain knowledge and languages, semantic heterogeneity is pervasive among different KBs. This problem has been widely discussed and investigated in the knowledge and data engineering areas (Euzenat and Shvaiko 2013). Some of the most obvious kinds of heterogeneity among different KBs are as follows: (i) Synonyms, which

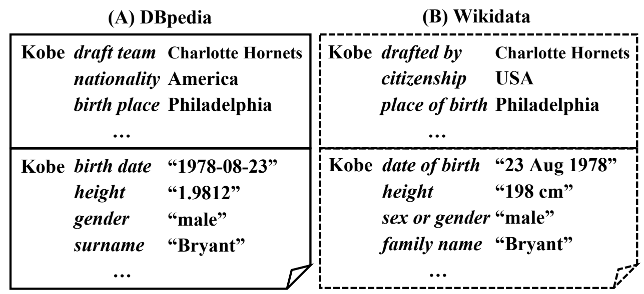

Figure 1: An example of the famous basketball player *Kobe Bryant* in DBpedia and its aligned entity in Wikidata.

occur when a term or phrase means exactly or nearly the same as another. (ii) Homonyms, i.e., the same term has different meanings. (iii) Various data formats. KBs may use different units of measurement and numerical precisions. (iv) Multilingualism. The natural languages used in KBs may be different, which leads to a great deal of heterogeneity. For example, Figure 1 shows two entities in DBpedia and Wikidata that refer to the basketball player *Kobe Bryant* (we call them *aligned entities* in this paper). This example demonstrates several types of semantic heterogeneity such as synonymous attributes (*surname* vs. *family name*) and values with different data formats ("1.9812" vs. "198 cm"). Additionally, the name *Kobe* is homonymous as it may refer to a Japanese city.

The aforementioned heterogeneity imposes many obstacles to KB integration. Existing approaches leverage various aspects of KBs (e.g., textual descriptions) and external resources (e.g., lexicons, Wikipedia links) (Euzenat and Shvaiko 2013; Suchanek, Abiteboul, and Senellart 2012; Wang, Li, and Tang 2013) to overcome the heterogeneity. However, entity descriptions are not always adequate enough and preparing external resources is often costly. In order to overcome these limitations, the embedding-based approaches (Chen et al. 2017; Zhu et al. 2017; Sun, Hu, and Li 2017) have been proposed recently. They follow the popular knowledge representation learning methods (Bordes et al. 2013; Wang et al. 2014; Lin et al. 2015b) and leverage KB embeddings to exploit the structure information of KBs for entity alignment. However, these existing approaches only focus on entity alignment but ignore relation and attribute alignment. Given that relations and attributes are essential ingredients of KBs, we speculate that relation and attribute alignment and entity alignment can reinforce each other. In this paper, we call the general problem of aligning entities, relations and attributes in different KBs as *KB alignment*.

We propose a novel embedding-based approach to KB alignment. It learns embeddings for entities, relations and attributes in a unified space. Each pair of aligned entities, relations or attributes is expected to be embedded closely. Our approach is motivated by the observation that two alignable KBs usually share a number of common facts, which are independent of their heterogeneity. Observe Figure 1 again. Entities for *Kobe Bryant* in DBpedia and Wikidata are very likely aligned as they share similar relation facts, and their attributes are highly related (i.e., these attributes all describe a person).

Specifically, our approach consists of two key modules:

- Heterogeneous Knowledge Base Embedding (HKBE) encodes all the entities, relations and attributes from heterogeneous KBs into a unified vector space in which the aligned ones are expected to lie closely. HKBE considers not only relations between entities but also their attributes. This enables our approach to integrally align entities, relations and attributes.
- Dynamic Alignment (DA) helps refine the embeddings learned by HKBE to enhance alignment. It dynamically enhances the similarities between potentially-aligned entities (i.e., the most possible entity alignment) found in the iterative learning process of HKBE, while weakening the similarities with other entities that are unlikely to form alignment.

We evaluate our approach on three pairs of heterogeneous datasets from DBpedia and Wikidata. The experimental results show that our approach can accurately discover entity, relation and attribute alignment and significantly outperforms the state-of-the-art embedding-based approaches.

Problem Modeling and Proposed Framework

We first give our definitions of KBs and KB alignment.

Definition 1 *(Knowledge Base) We define a KB as a 6-tuple* (E, R, A, L, T_r, T_a)*, where* E, R, A *and* L *denote the sets of entities, relations, attributes and literals, respectively.* $T_r \subseteq E \times R \times E$ *denotes the set of relation triples.* $T_a \subseteq E \times A \times L$ *denotes the set of attribute triples.*

Different KBs often use heterogeneous relations and attributes to describe the same entity. Thus, we define a more general version of KB alignment as follows.

Definition 2 *(Knowledge Base Alignment) Let* KB_1, KB_2 *denote two KBs. KB alignment is to find semantically-equivalent entities, relations and attributes in the two KBs.*

Following the duplicate-free assumption on KBs (Zhang, Rubinstein, and Gemmell 2015; Megdiche, Teste, and dos Santos 2016), we particularly address *one-to-one* alignment in this paper, e.g., one entity in KB_1 can at most align with one entity in KB_2. Already-aligned entities, relations and attributes such as those made possible by the Linking Open Data movement can serve as a bridge between KBs. We call them *seed alignment* in this paper.

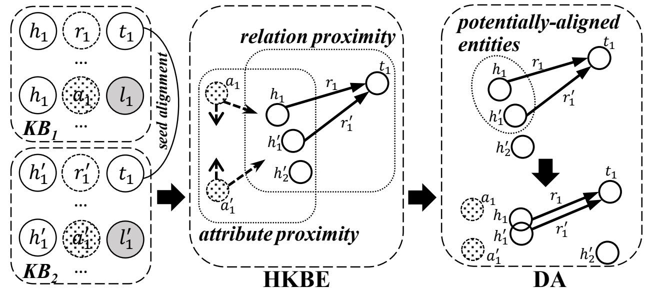

Figure 2: The proposed framework. Solid, dotted, shadowed and gray circles refer to entities, relations, attribute and literals, respectively. In the vector space, entities and attributes can be regarded as points, while relations can be regarded as solid arrows. Dotted arrows indicate attribute proximity.

To tackle the problem of KB alignment, we propose an embedding-based approach. It takes as input two KBs and some seed alignment, and uses HKBE and DA to learn the embeddings of entities, relations and attributes for aligning them. Figure 2 illustrates the framework of our approach.

HKBE HKBE maps all the entities, relations and attributes of two KBs into a unified low-dimensional vector space $\mathbb{R}^d$, so that the similarities between these elements can be measured via the distance between their embeddings.

HKBE learns embeddings from relation triples and attribute triples. We propose *relation proximity* and *attribute proximity* to encode them, respectively. Relation triples in a KB can be modeled as a relation graph, in which each vertex denotes an entity and each edge denotes a relation. Relation proximity is designed based on the intuition that entities or relations in two relation graphs are likely aligned when their adjacent nodes are close to each other. Thus, HKBE lets each pair of seed alignment share the same embedding and learns the embeddings of entities and relations to model the structural information of relation graphs, so that potentially-aligned elements can stay close in the unified vector space. For example, h_1 is adjacent to h_1' in Figure 2 because they have similar relations to a common entity t_1.

On the other hand, attribute proximity explores two aspects of attribute triples: (i) a group of attributes are often used together to constitute a facet for a type of entities (Gong et al. 2016); and (ii) similar entities should have related attributes. The first aspect is pertinent to the relatedness among attributes. We expect attributes constituting the same facet to be embedded in close vicinity as they are usually related. The second is about associations between entities and their attributes. We expect entities and their attributes to be embedded closely. The two aspects are denoted by dotted open and dovetail arrows in Figure 2, respectively. For example, based on attribute proximity, *Kobe* with attributes {*gender, birth date, height*} should stay far away from *Kobe* with attributes {*area, postal code, population*}. The two groups of attributes have no relatedness, therefore their associated entities should be dissimilar. Note that in this framework we do not consider attribute values due to their tremendous heterogeneity and complexity.

DA During the iterative learning process of HKBE, the embeddings of truly-aligned entities, relations and attributes are expected to gradually approach each other, while the embeddings of non-aligned ones should move away from each other. To strengthen true alignment and eliminate "noises" and outliers, we introduce DA, which dynamically finds potentially-aligned entities based on their current embeddings learned by HKBE and leverages them as supervision for guiding the succeeding embedding learning process. Specifically, we incorporate the one-to-one alignment constraint to select the potentially-aligned entities with the highest overall similarity. For example, DA would choose $h1$ and h'_1 in Figure 2 as aligned, due to their adjacency and similar relations to seed alignment. DA then pulls them closer. Other non-aligned entities for $h1$, such as h'_2, would be pushed away. Note that the update of entity embeddings also refines relation and attribute embeddings due to relation and attribute proximity.

Together, HKBE and DA dynamically and integrally optimize embeddings of entities, relations and attributes while preserving their relation proximity and attribute proximity.

Dynamic Alignment via HKBE

The framework of our approach mainly includes HKBE and DA. Thus, we combine their objective functions to define the following overall objective:

$$\mathcal{O} = \mathcal{O}_{hkbe} + \mathcal{O}_{da}, \tag{1}$$

where $\mathcal{O}_{hkbe}$ and $\mathcal{O}_{da}$ denote the objective functions of HKBE and DA, respectively.

Heterogeneous Knowledge Base Embedding

HKBE consists of two modules: (i) *Relation Proximity Embedding* (RPE), which models the relation proximity of entities and relations in KBs, and (ii) *Attribute Proximity Embedding* (APE), which encodes the relatedness between attributes and their associations with entities. Thus, we define the joint objective function of HKBE as follows:

$$\mathcal{O}_{hkbe} = \mathcal{O}_{rpe} + \mathcal{O}_{ape}, \tag{2}$$

where $\mathcal{O}_{rpe}$ and $\mathcal{O}_{ape}$ denote the objective functions of RPE and APE, respectively. It is worth mentioning that each pair in the seed alignment shares the same embedding in the unified vector space.

Relation Proximity Embedding To model the relation proximity of two relation graphs, we aim to align them with each pair in the seed alignment sharing the same embedding. We denote by $\mathcal{P}(r|h,t)$ the probability that (h,r,t) is a real relation triple and define it as follows:

$$\mathcal{P}(r|h,t) = \sigma\big(f(h,r,t)\big), \tag{3}$$

where $f()$ is the score function to measure the plausibility of a relation triple and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function. Correspondingly, $1 - \mathcal{P}(r|h,t) = \sigma\big(-f(h,r,t)\big)$ is the probability that (h,r,t) is a fake triple, which is generated by replacing the heads or tails of relation triples with random entities (Bordes et al. 2013).

To encode the relation triples, we borrow the idea from the translation-based embedding model TransE (Bordes et al. 2013) and regard a relation as a translation vector from its head to tail. The score function is defined as follows:

$$f(h,r,t) = -\|\vec{v}(h) + \vec{v}(r) - \vec{v}(t)\|_2^2, \tag{4}$$

where $\vec{v}()$ denotes the vector representation of a given entity, relation or attribute. We prefer higher scores for real triples and lower scores for fake ones.

To model both real and fake relation triples, we use the maximum likelihood estimation to parameterize the embeddings. We minimize the following negative log-likelihood function:

$$\begin{aligned}\mathcal{O}_{rpe} = &-\sum_{(h,r,t)\in T_r} \log\sigma\big(f(h,r,t)\big)\\ &-\sum_{(h',r',t')\in T'_r} \alpha\log\sigma\big(-f(h',r',t')\big),\end{aligned} \tag{5}$$

where T'_r is the set of fake triples. The first term models the real triples while the second term models the fake ones. We introduce a hyper-parameter α to balance the influence of real and fake triples.

Attribute Proximity Embedding To model the attribute proximity, we extend Skip-gram (Mikolov et al. 2013a) and predict both related attributes and associated entities given an attribute. Thus, we minimizes the following objective:

$$\begin{aligned}\mathcal{O}_{ape} = -\sum_{a_i\in A}\Big(&\sum_{a_j\in c_a(a_i)} \log\mathcal{P}(a_j|a_i)\\ &+\sum_{e\in c_e(a_i)} \beta\log\mathcal{P}(e|a_i)\Big),\end{aligned} \tag{6}$$

where $c_a(a_i)$ and $c_e(a_i)$ denote the sets of related attributes and associated entities of a_i, respectively. β is a hyper-parameter to balance the two terms. Given an attribute a_i, the first term models the probability to predict its related attributes and the second models its associations with entities. Formally, the conditional probability $\mathcal{P}(a_j|a_i)$ is formulated using softmax as follows:

$$\mathcal{P}(a_j|a_i) = \frac{\exp\big(\vec{v}(a_j)\cdot\vec{v}(a_i)\big)}{\sum_{a_k\in A}\exp\big(\vec{v}(a_k)\cdot\vec{v}(a_i)\big)}. \tag{7}$$

It is impractical to directly solve Eq. (7) because computing the sum in the denominator to normalize the probability is costly. Thus, we adopt the negative sampling method (Mikolov et al. 2013b) to approximate the log probability of softmax and redefine $\log\mathcal{P}(a_j|a_i)$ as follows:

$$\begin{aligned}\log\mathcal{P}(a_j|a_i) &= \log\sigma\big(\vec{v}(a_j)\cdot\vec{v}(a_i)\big)\\ &+\sum_{n=1}^{m}\mathbb{E}_{a_n\sim D_n(a)}\big[\log\sigma\big(-\vec{v}(a_n)\cdot\vec{v}(a_i)\big)\big],\end{aligned} \tag{8}$$

where a_n is a non-related attribute for a_i, which is sampled according to a noisy distribution $D_n(a) \propto d_a^{3/4}$ and d_a is the number of times that attribute a occurs. m denotes the number of non-related attributes sampled for each attribute. Analogously, $\mathcal{P}(e|a_i)$ can be solved in the same way.

Dynamic Alignment

DA aims to find potentially-aligned entities during the learning process of HKBE and use them to refine embeddings for better alignment. We use subscript $[i]$ to mark elements in the i-th epoch of training. For instance, the embedding of entity e in the i-th epoch is denoted by $\vec{v}_{[i]}(e)$. The similarity between two entity vectors is calculated using cosine: $s_{[i]}(e_1, e_2) = \cos\big(\vec{v}_{[i]}(e_1), \vec{v}_{[i]}(e_2)\big) = \vec{v}_{[i]}(e_1) \cdot \vec{v}_{[i]}(e_2)$. Furthermore, we define an operator $n(e_1, e_2)$ to count the number of common seed aligned entities connected by e_1 and e_2 first, and then its values are normalized to $[0, 1]$. To synthesize the vector similarity and the influence from seed alignment, we define a combined similarity measure $\mu_{[i]}()$ as follows:

$$\mu_{[i]}(e_1, e_2) = s_{[i]}(e_1, e_2) + \rho \cdot n(e_1, e_2), \tag{9}$$

where ρ is a small hyper-parameter. $\mu_{[i]}(e_1, e_2)$ indicates how likely e_1 and e_2 are aligned in the i-th epoch, which is mainly dependent on the vector similarity $s_{[i]}(e_1, e_2)$ and heightened based on $n(e_1, e_2)$. To accelerate the selection process, we employ a k-nearest neighbor strategy that requires the potentially-aligned counterpart for an entity to be in its k-nearest neighbors. Let $t^k_{[i]}()$ indicate whether an entity is in the k-nearest neighbors of another, where $t^k_{[i]}(e_1, e_2) = 1$ if e_2 is one of the k-nearest neighbors of e_1, and 0 otherwise. We redefine Eq. (9) below:

$$\mu_{[i]}(e_1, e_2) = t^k_{[i]}(e_1, e_2)\big(s_{[i]}(e_1, e_2) + \rho \cdot n(e_1, e_2)\big). \tag{10}$$

Let $\phi_{[i]}()$ denote whether an entity pair is selected, where $\phi_{[i]}(e_1, e_2) = 1$ if e_1 and e_2 are chosen as two potentially-aligned entities, and 0 otherwise. To reflect the one-to-one alignment constraint and guarantee the quality of selected potentially-aligned entities, we optimize the selection process in the i-th epoch of training as follows:

$$\begin{aligned} &\max \sum_{e_1 \in \hat{E}_1} \sum_{e_2 \in \hat{E}_2} \phi_{[i]}(e_1, e_2) \cdot \mu_{[i]}(e_1, e_2), \\ &\text{s.t.} \sum_{e'_2 \in \hat{E}_2} \phi_{[i]}(e_1, e'_2) \le 1, \sum_{e'_1 \in \hat{E}_1} \phi_{[i]}(e'_1, e_2) \le 1, \\ &\quad \mu_{[i]}(e_1, e_2) > \tau, \end{aligned} \tag{11}$$

where $\hat{E}_1$ and $\hat{E}_2$ denote the sets of entities in KB_1 and KB_2 to be aligned, respectively.

After solving Eq. (11), we obtain a potential entity alignment $M_{[i]} = \{(e_1, e_2) \in \hat{E}_1 \times \hat{E}_2 | \phi_{[i]}(e_1, e_2) = 1\}$. Then, we generate the supervision triples $\hat{T}_{r[i]}$. For each $(e_1, e_2) \in M_{[i]}$, we create new triples simply by using e_1 in KB_1 to replace e_2 in KB_2, and vice versa. Thus, $\hat{T}_{r[i]} = \{(e_2, r, t) | (e_1, r, t) \in T_{r1})\} \cup \{(e_1, r, t) | (e_2, r, t) \in T_{r2})\}$, where T_{r1} and T_{r2} represent the sets of relation triples in KB_1 and KB_2, respectively. Additionally, to eliminate "noises", we randomly sample fake alignment and create new fake triples $\hat{T}'_{r[i]}$ in the same way.

Finally, we reuse Eq. (5) to encode new triples and fake triples:

$$\mathcal{O}_{da} = \mathcal{O}_{rpe}[T_r := \hat{T}_{r[i]}, T'_r := \hat{T}'_{r[i]}], \tag{12}$$

which replaces T_r and T'_r in $\mathcal{O}_{rpe}$ with $\hat{T}_{r[i]}$ and $\hat{T}'_{r[i]}$, respectively. Please note that it is straightforward to expand DA to find potentially-aligned relations or attributes. For example, it is recipient that aligned attributes should have the same range type, and we add this constraint when performing attribute alignment.

Implementation Details

Training We initialize the embeddings of entities, relations and attributes randomly based on a truncated normal distribution, and use the gradient descent optimization algorithm Ada-Grad (Duchi, Hazan, and Singer 2011) to optimize Eq. (1). The length of all vectors is restrained to 1. Instead of directly optimizing $\mathcal{O}_{hkbe}$, our training process involves two optimizers to minimize $\mathcal{O}_{rpe}$ and $\mathcal{O}_{ape}$ independently. At each epoch, the two optimizers are executed alternately. $\mathcal{O}_{hkbe}$ and $\mathcal{O}_{da}$ are also optimized separately.

Maximum Weighted Graph Matching (MWGM) We reduce selecting potentially-aligned entities in Eq. (11) to the MWGM problem. We first extract entity pairs that satisfy their similarity $\mu_{[i]}(e_1, e_2) > \tau$, and then construct a bipartite graph whose nodes represent entities and edges have weights representing the similarities between nodes. Thus, selecting potentially-aligned entities with the maximum similarity is transformed to finding disjoint edges with the maximum weights in MWGM problem.

Complexity Analysis The number of parameters in our entire model is $d(N_e, N_r, N_a)$, where N_e, N_r, N_a denote the numbers of all the entities, relations and attributes to be embedded, respectively. The time complexity of MWGM can be reduced to linear time $O(N'_v + N'_e)$ using the heuristic algorithm in (Abou-Rjeili and Karypis 2006), where N'_v and N'_e denotes the numbers of nodes and edges in the bipartite graph, respectively, during the current round of DA.

Experiments

We used Tensorflow to develop our approach, called AREA (the inverted order of Alignment of Entities, Relations and Attributes), and evaluated it on three KB alignment tasks: entity alignment, relation alignment and attribute alignment. The source code and datasets will be available online.

Datasets

We built three pairs of heterogeneous datasets using DBpedia (2016-04) and Wikidata, referred to as $\text{DBP}_{\text{EN}} \rightarrow \text{DBP}_{\text{FR}}$, $\text{DBP}_{\text{EN}} \rightarrow \text{WD}$ and $\text{DBP}_{\text{FR}} \rightarrow \text{WD}$, representing three different categories of heterogeneity. We can imagine that $\text{DBP}_{\text{FR}} \rightarrow \text{WD}$ is the hardest. DBpedia is a large-scale, multi-lingual KB, providing rich entity links (i.e., alignment) from its English version to other languages and also to other KBs such as Wikidata. Taking $\text{DBP}_{\text{EN}} \rightarrow \text{WD}$ for example, we first randomly picked 15 thousand entity alignment from the English version of DBpedia to Wikidata. Each entity in the alignment involves at least 5 triples. Then we extracted the relation and attribute triples only involving the entities in these 15K alignment. Based on the reference relation and attribute alignment provided in the DBpedia ontology, we generated

Table 1: Statistics of the datasets

	$DBP_{EN} \rightarrow DBP_{FR}$		$DBP_{EN} \rightarrow WD$		$DBP_{FR} \rightarrow WD$	
	DBP_{EN}	DBP_{FR}	DBP_{EN}	WD	DBP_{FR}	WD
Entities	15,000	15,000	15,000	15,000	15,000	15,000
Relations	199	167	216	148	155	125
Attributes	184	241	225	516	267	483
Rel. triples	59,568	54,131	62,168	73,545	82,281	76,570
Attr. triples	52,260	59,965	74,118	132,135	80,943	126,610
Ent. align.	15,000		15,000		15,000	
Rel. align.	97		80		47	
Attr. align.	84		50		32	

80 relation alignment and 50 attribute alignment as the gold standards. The other two datasets were built in the same way.

The statistics of the datasets are listed in Table 1. We can see that the numbers of relations and attributes between the datasets vary a lot, which reflects the great heterogeneity of them. This also leads to the relation and attribute alignment are few as well. For example, even though DBP_{EN} and DBP_{FR} use the same ontology, there are not many relation and attribute alignment between them.

Comparative Approaches

For the entity and relation alignment tasks, we chose three latest embedding-based approaches: MTransE (Chen et al. 2017), IPTransE (Zhu et al. 2017) and JAPE (Sun, Hu, and Li 2017). All of them focus on entity alignment via knowledge embeddings and are capable of learning relation embeddings, which allowed us to carry out relation alignment with them. MTransE contains five variants in its alignment model, where the fourth performs best according to the experiments of its authors. Thus, we chose this variant to represent MTransE. For IPTransE, we used its best configuration including the parameter sharing and iterative alignment modules. Note that IPTransE relies on PTransE (Lin et al. 2015a), which incorporates additional reverse triples and relation paths, while the other methods do not. We tried our best to implement IPTransE, since its source code has not been released so far. We used the full model of JAPE.

For attribute alignment, we employed two popular word embedding models, CBOW and Skip-gram (Mikolov et al. 2013a), to learn attribute embeddings, because MTransE and IPTransE do not deal with attributes at present. Skip-gram predicts related attributes given one attribute, while CBOW predicts the current attribute given its related attributes. We still used JAPE because it can model attribute relatedness.

Experiment Settings

We used 20% of the gold standards as the seed alignment, while left the remaining as the testing data (i.e., the entity, relation and attribute alignment to be discovered). Given a source entity, relation or attribute, finding its aligned counterpart was achieved by searching its nearest neighbors in the unified vector space. We employed cosine similarity to measure the distance between embeddings.

To train AREA, we tuned various parameter values and a good configuration is $\alpha = 1$ (for HKBE), $\alpha = 0.1$ (for DA), $\beta = 0.01, m = 1, \rho = 0.2$ and $\tau = 0.6$. We sampled five fake triples for each real triple. We chose $k = 10$ in DA to achieve a balance between accuracy and efficiency, and will also report the results of other choices shortly. One epoch of DA was performed when training 10 epochs of HKBE. The learning rate is set to 0.01 and the training takes $1,000$ epochs. Parameter settings for the comparative approaches followed their papers. For all the approaches, vector dimensions were set to 75 for a fair comparison.

Following the conventions in (Toutanova and Chen 2015; Chen et al. 2017; Zhu et al. 2017; Sun, Hu, and Li 2017), we chose $Hits@n$ (abbr. $H@n$ in tables), mean rank (MR) and mean reciprocal rank (MRR) as our evaluation metrics. $Hits@n$ measures the proportion of correctly-aligned ones ranked in the top n, while MR calculates the mean of these ranks. MRR is the average of the reciprocal ranks of results. A higher $Hits@n$, a lower MR and a higher MRR indicate better performance.

Results and Discussion

Entity Alignment Table 2 lists the results of entity alignment. We can see that the proposed approach AREA significantly outperformed MTransE, IPTransE and JAPE, due to the fact that AREA captured sufficient information, i.e., the relation and attribute proximity, for entity alignment, while MTransE and IPTransE only leveraged relation triples. JAPE also learned the embeddings for attributes, however, it modeled entities and attributes in different vector spaces, which broke their semantic associations. As IPTransE leveraged additional reverse triples and relation paths, its performance measured by MR is relatively superior than $Hits@n$.

It is worth noting that our DA module contributed significantly. When adding DA, the results on entity alignment improved a lot, especially on $Hits@1$. This indicates the practical usefulness of our approach, as a higher $Hits@1$ reflects the key effectiveness of entity alignment. The great performance of DA is due to its capability to accurately capture alignment clues. We believe that DA can be a universal enhancement for the embedding-based alignment models.

Moreover, we found that the results on the three datasets, from $DBP_{EN} \rightarrow DBP_{FR}$, $DBP_{EN} \rightarrow WD$ to $DBP_{FR} \rightarrow WD$, decline gradually. This is consistent with our intuitions, as the semantic heterogeneity in these three datasets increases accordingly. However, even on the most heterogeneous dataset $DBP_{FR} \rightarrow WD$, AREA still achieved acceptable results, e.g., $Hits@1$ reaches 42.78, which demonstrated its feasibility.

Relation Alignment The results of relation alignment are listed in Table 3. We can observe that all the approaches achieved acceptable results as they are all able to encode the structural information of KBs. It is impressive that IPTransE performed magnificently on relation alignment, especially in terms of MR, even though its results on entity alignment is not outstanding. We think this is due to the extra reverse triples and relation paths added in IPTransE, which provide more information to encode the structure of KBs. For AREA. it obtained superior results on all the datasets, especially in terms of $Hits@1$ and MRR, which indicated the stable performance of AREA on relation alignment.

Table 2: Result comparison on entity alignment

	$\text{DBP}_{\text{EN}}\rightarrow\text{DBP}_{\text{FR}}$					$\text{DBP}_{\text{EN}}\rightarrow\text{WD}$					$\text{DBP}_{\text{FR}}\rightarrow\text{WD}$				
	$H@1$	$H@5$	$H@10$	MR	MRR	$H@1$	$H@5$	$H@10$	MR	MRR	$H@1$	$H@5$	$H@10$	MR	MRR
MTransE	26.05	44.61	51.80	429.91	0.348	21.32	38.31	45.58	322.59	0.295	17.60	37.87	47.10	505.86	0.272
IPTransE	31.76	62.16	74.94	93.77	0.456	29.02	53.87	63.82	99.23	0.407	15.82	33.94	42.44	237.48	0.249
JAPE	39.24	65.86	75.38	128.26	0.513	29.12	54.44	64.55	100.59	0.408	20.88	43.40	51.88	373.54	0.311
HKBE	41.42	68.82	79.16	80.70	0.538	33.98	60.92	71.39	70.56	0.464	20.95	42.93	53.17	231.34	0.316
AERA	**63.64**	**81.82**	**86.38**	**51.76**	**0.719**	**56.68**	**75.42**	**80.16**	**56.03**	**0.653**	**42.78**	**64.44**	**71.51**	**186.20**	**0.527**

A higher $H@n$, a lower MR and a higher MRR indicate a better result. The best results are marked in bold. The same to the following.

Table 3: Result comparison on relation alignment

	$\text{DBP}_{\text{EN}}\rightarrow\text{DBP}_{\text{FR}}$				$\text{DBP}_{\text{EN}}\rightarrow\text{WD}$				$\text{DBP}_{\text{FR}}\rightarrow\text{WD}$			
	$H@1$	$H@5$	MR	MRR	$H@1$	$H@5$	MR	MRR	$H@1$	$H@5$	MR	MRR
MTransE	43.59	62.82	8.97	0.541	39.06	64.06	12.08	0.504	23.68	47.37	9.45	0.374
IPTransE	58.97	84.62	4.26	0.693	48.44	70.31	**7.06**	0.603	36.84	55.26	8.53	0.470
JAPE	56.41	76.92	5.99	0.670	45.31	65.62	11.88	0.563	26.32	60.53	7.26	0.417
HKBE	57.69	87.18	4.08	0.698	54.69	68.75	7.61	0.626	42.11	63.16	6.90	0.534
AREA	**69.23**	**91.03**	**3.39**	**0.783**	**60.94**	**76.56**	9.19	**0.683**	**57.89**	**76.32**	**5.45**	**0.689**

Table 4: Result comparison on attribute alignment

	$\text{DBP}_{\text{EN}}\rightarrow\text{DBP}_{\text{FR}}$				$\text{DBP}_{\text{EN}}\rightarrow\text{WD}$				$\text{DBP}_{\text{FR}}\rightarrow\text{WD}$			
	$H@1$	$H@5$	MR	MRR	$H@1$	$H@5$	MR	MRR	$H@1$	$H@5$	MR	MRR
CBOW	2.94	13.24	30.62	0.097	0.00	10.00	21.50	0.091	7.69	19.23	13.08	0.172
Skip-gram	2.94	7.35	32.60	0.081	0.00	12.50	19.93	0.091	3.85	11.54	13.65	0.137
JAPE	19.12	58.82	11.82	0.358	10.00	47.50	10.45	0.280	11.54	46.15	9.19	0.300
HKBE	39.71	72.06	9.49	0.528	37.50	**72.50**	6.98	0.546	19.23	61.54	7.54	0.380
AREA	**44.12**	**76.47**	**6.07**	**0.601**	**42.50**	**72.50**	**6.28**	**0.566**	**26.92**	**65.38**	**7.12**	**0.436**

Attribute Alignment Table 4 shows the results of our approach on attribute alignment. Two baselines methods, CBOW and Skip-gram, failed to obtain desired results. The reason may be that the relatedness of attributes is relatively simple compared to the complicated dependence between words. Duo to encoding the range type information, the performance of JAPE improved largely. AREA achieved the best results by encoding both attribute relatedness and their associations with entities.

Analysis of k-Nearest Neighbors in DA Figure 3 illustrates the performance change of entity alignment with varied k. We carried out $k = 5, 10, 20$ and 50. For comparison, we also disabled the k-nearest neighbor strategy (marked to as "W/O" in the figure). We observed that choosing smaller k brought limited influence for $Hits@n$ but significant acceleration for running time. For example, on DBP_{EN} →WD, $Hits@1$ under $k = 5, 10, 20, 50$ and without k is 55.80, 56.68, 57.67, 57.96 and 56.83, respectively, whereas the average running time per DA round is 18.59s, 21.91s, 27.31s, 42.81s and 140.63s. Therefore, the k-nearest neighbor strategy can save much running time without losing much accuracy. To achieve the balance, we used $k = 10$.

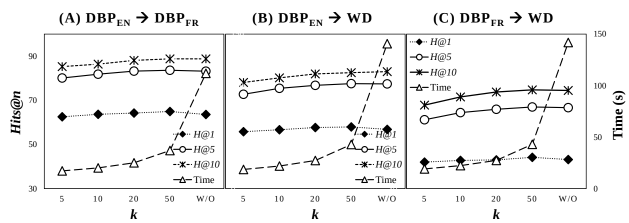

Figure 3: $Hits@n$ and running time on entity alignment w.r.t. k-nearest neighbors (or without k)

Sensitivity to Proportion of Seed Alignment We tested the proportion of seed alignment from 10% to 40% with step 10%. Figure 4 illustrates the change of $Hits@n$ with different proportion. With the increase of the proportion, the results on the three datasets became better as expected, because more seed alignment can provide more information to align two KBs. When only using 10% of the gold standard as the seed alignment, AREA still achieved satisfactory results. For example, $Hits@5$ on the three datasets reached $64.23, 61.39$ and 44.16. We also found that the differences between $Hits@5$ and $Hits@10$ are very smaller for all the proportions. This implies that AREA ranked correct entities highly, which has also been verified by the high MRR in Table 2.

Robustness at Larger Scale To assess the scalability of AREA, we build three larger datasets (100K) following the same method as aforementioned. Each dataset has 100 thousand randomly-picked entity alignment. We used the same experiment settings as 15K. Due to space limitation, we only reported the results on $\text{DBP}_{\text{EN}}\rightarrow$WD (100K) in Table 5. Our approach AREA still obtained the best results on the three alignment tasks, which indicated the scalability and stability of AREA. Note that results on 100K decreased compared to that on 15K. We think it is mainly because that $\text{DBP}_{\text{EN}}\rightarrow$WD (100K) is more sparse. Similar results and conclusions stand for the other two datasets.

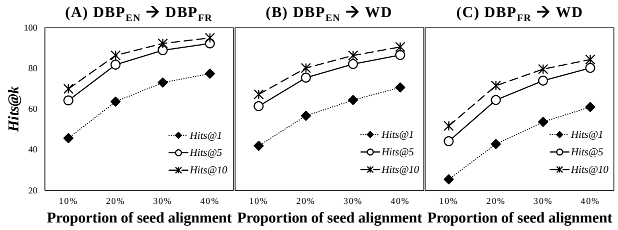

Figure 4: $Hits@n$ on entity alignment w.r.t. proportion of seed alignment

Table 5: Result comparison on DBP$_{EN}$→WD (100K)

	Ent. alignment		Rel. alignment		Attr. alignment	
	$H@10$	MRR	$H@5$	MRR	$H@5$	MRR
MTransE	32.18	0.203	64.56	0.537	-	-
IPTransE	43.40	0.274	72.15	0.618	-	-
JAPE	39.81	0.255	73.42	0.633	54.55	0.383
CBOW	-	-	-	-	11.36	0.120
Skip-gram	-	-	-	-	11.36	0.086
HKBE	55.35	0.385	72.15	0.629	61.36	0.474
AREA	**67.28**	**0.536**	**74.68**	**0.659**	**65.91**	**0.504**

Related Work

We divide the related work into two subfields: KB embedding and KB alignment, which we discuss below.

KB Embedding

Recently, learning embeddings for KBs has demonstrated its effectiveness in modeling the structural information of KBs. TransE (Bordes et al. 2013) interprets a relation vector as the translation from its head entity vector to tail entity vector. This translation-based KB embedding model has shown its superiority for KB completion and is followed by many studies. For example, TransH (Wang et al. 2014), TransR (Lin et al. 2015b) and STransE (Nguyen et al. 2016) were proposed to improve TransE on modeling multi-mapping relations. Additionally, there exist some non-translation-based approaches to KB embedding, such as (Bordes et al. 2011; Nickel, Tresp, and Kriegel 2011; Socher et al. 2013).

Additionally, several methods leverage extra knowledge in KBs to improve embedding. PTransE (Lin et al. 2015a) incorporates reverse triples and relation paths. KR-EAR (Lin, Liu, and Sun 2016) distinguishes attributes from relations, and learns attribute embeddings by modeling attribute correlations. Attributes used in KR-EAR is not the same as ours and their attributes are essentially a kind of categorical relations, e.g., *nationality*. Besides, type information and local structure of entities were explored by (Krompaß, Baier, and Tresp 2015; Ristoski and Paulheim 2016).

KB Alignment

With the increasing number and scale of KBs, KB alignment has attracted a lot of attentions. PARIS (Suchanek, Abiteboul, and Senellart 2012) is a probability-based approach for joint alignment of instance, relations and classes. It models how two instance are likely to be aligned based on their compatible neighbors and attribute values. SiGMa (Lacoste-Julien et al. 2013) is an iterative propagation algorithm for entity alignment. It leverages both the structural information of relation graph and the flexible similarity measures between entity attributes. Due to the fact that PARIS and SiGMa rely on literal similarities, they would suffer from the semantic heterogeneity between large KBs. In terms of cross-lingual KB alignment, the work in (Wang, Li, and Tang 2013) leverages language-independent features (e.g., out/in-links) to find cross-lingual links between wiki KBs. In contrast, our approach is based on the KB embedding techniques and does not need external resources.

Recently, several embedding-based methods for entity alignment have been proposed. MTransE (Chen et al. 2017) adopts TransE to embed different KBs into different vector spaces, and learns semantic translation between KBs via five different alignment models. JAPE (Sun, Hu, and Li 2017) learns the embeddings for entities and relations of different KBs in a unified vector space. It also embeds attributes and leverages attribute embeddings to calculate entity similarities. However, it models entities and attributes in different vector spaces, which cuts off their semantic associations. None of them involves the dynamic part to refine embeddings. IPTransE (Zhu et al. 2017) is a complex framework for entity alignment. It employs PTransE (Lin et al. 2015a) to embed single KB and proposes three modules (translation-based, linear transformation and parameter sharing) for jointly embedding different KBs. It also considers using newly-aligned entities to update embeddings iteratively. However, it finds newly-aligned entities only based on vector distances, which are cumulative in different rounds. This process is error-prone. IPTransE assumes that relation alignment is known, which is a strong assumption. Different from these methods, our approach encodes entities, relations and attributes in a unified vector space to integrally align them.

Conclusion and Future Work

The main contributions of this paper are summarized below.

- We defined the general problem of KB alignment and proposed an embedding-based approach for aligning entities, relations and attributes integrally.
- To resolve the semantic heterogeneity between different KBs, we designed HKBE to encode all entities, relations, attributes into a unified vector space while preserving their relation proximity and attribute proximity.
- We presented DA to leverage seed and potentially-aligned entities for enhancing the learning process dynamically. The k-nearest neighbor selection strategy was employed to improve efficiency.
- We built three pairs of heterogeneous datasets from DBpedia and Wikidata. Our experimental results showed that our approach significantly outperformed the state-of-the-art embedding-based methods.

For future work, we plan to study the cross-lingual word embedding for attribute values. We also want to incorporate the recurrent neural networks to model the complex structures of KBs.

References

Abou-Rjeili, A., and Karypis, G. 2006. Multilevel algorithms for partitioning power-law graphs. In *Proc. of the 20th International Parallel and Distributed Processing Symposium*, 124. Rhodes Island, Greece: IEEE.

Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In *Proc. of the 25th AAAI Conference on Artificial Intelligence*, 301–306. San Francisco, USA: AAAI.

Bordes, A.; Usunier, N.; Garcia-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems*, 2787–2795. Lake Tahoe, USA: Curran Associates Inc.

Chen, M.; Tian, Y.; Yang, M.; and Zaniolo, C. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proc. of the 26th International Joint Conference on Artificial Intelligence*, 1511–1517. Melbourne, Australia: AAAI.

Duchi, J. C.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12(7):2121–2159.

Euzenat, J., and Shvaiko, P. 2013. *Ontology matching*. Springer, second edition.

Gong, S.; Li, H.; Hu, W.; and Qu, Y. 2016. An empirical study on property clustering in linked data. In *Proc. of the 6th Joint International Semantic Technology Conference*, 67–82. Singapore: Springer.

Krompaß, D.; Baier, S.; and Tresp, V. 2015. Type-constrained representation learning in knowledge graphs. In *Proc. of the 14th International Semantic Web Conference*, 640–655. Bethlehem, USA: Springer.

Lacoste-Julien, S.; Palla, K.; Davies, A.; Kasneci, G.; Graepel, T.; and Ghahramani, Z. 2013. SiGMa: Simple greedy matching for aligning large knowledge bases. In *Proc. of the 19th ACM International Conference on Knowledge Discovery and Data Mining*, 572–580. Chicago, USA: ACM.

Lin, Y.; Liu, Z.; Luan, H.-B.; Sun, M.; Rao, S.; and Liu, S. 2015a. Modeling relation paths for representation learning of knowledge bases. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics*, 705–714. Beijing, China: ACL.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015b. Learning entity and relation embeddings for knowledge graph completion. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, 2181–2187. Austin, USA: AAAI.

Lin, Y.; Liu, Z.; and Sun, M. 2016. Knowledge representation learning with entities, attributes and relations. In *Proc. of the 25th International Joint Conference on Artificial Intelligence*, 2866–2872. New York, USA: AAAI.

Megdiche, I.; Teste, O.; and dos Santos, C. T. 2016. An extensible linear approach for holistic ontology matching. In *Proc. of the 15th International Semantic Web Conference*, 393–410. Kobe, Japan: Springer.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. In *Proc. of the 1st International Conference on Learning Representations*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems*, 3111–3119. Lake Tahoe, USA: Curran Associates Inc.

Nguyen, D. Q.; Sirts, K.; Qu, L.; and Johnson, M. 2016. STransE: A novel embedding model of entities and relationships in knowledge bases. In *Proc. of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 460–466. San Diego, USA: ACL.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proc. of the 28th International Conference on Machine Learning*, 809–816. Bellevue, USA: Omnipress.

Ristoski, P., and Paulheim, H. 2016. RDF2Vec: RDF graph embeddings for data mining. In *Proc. of the 15th International Semantic Web Conference*, 498–514. Kobe, Japan: Springer.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of the 27th Annual Conference on Neural Information Processing Systems*, 926–934. Lake Tahoe, USA: Curran Associates Inc.

Suchanek, F. M.; Abiteboul, S.; and Senellart, P. 2012. PARIS: Probabilistic alignment of relations, instances, and schema. *PVLDB* 5(3):157–168.

Sun, Z.; Hu, W.; and Li, C. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proc. of the 16th International Semantic Web Conference*. Vienna, Austria: Springer.

Toutanova, K., and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *Proc. of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, 57–66. Beijing, China: ACL.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proc. of the 28th AAAI Conference on Artificial Intelligence*, 1112–1119. Québec City, Canada: AAAI.

Wang, Z.; Li, J.; and Tang, J. 2013. Boosting cross-lingual knowledge linking via concept annotation. In *Proc. of 23rd International Joint Conference on Artificial Intelligence*, 2733–2739. Beijing, China: AAAI.

Zhang, D.; Rubinstein, B. I. P.; and Gemmell, J. 2015. Principled graph matching algorithms for integrating multiple data sources. *IEEE TKDE* 27(10):2784–2796.

Zhu, H.; Xie, R.; Liu, Z.; and Sun, M. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proc. of the 26th International Joint Conference on Artificial Intelligence*, 4258–4264. Melbourne, Australia: AAAI.