Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding

Zequn Sun[1], Wei Hu[1(✉)], and Chengkai Li[2]

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, China
zqsun.nju@gmail.com, whu@nju.edu.cn
[2] Department of Computer Science and Engineering,
University of Texas at Arlington, USA
cli@uta.edu

Abstract. Entity alignment is the task of finding entities in two knowledge bases (KBs) that represent the same real-world object. When facing KBs in different natural languages, conventional cross-lingual entity alignment methods rely on machine translation to eliminate the language barriers. These approaches often suffer from the uneven quality of translations between languages. While recent embedding-based techniques encode entities and relationships in KBs and do not need machine translation for cross-lingual entity alignment, a significant number of attributes remain largely unexplored. In this paper, we propose a joint attribute-preserving embedding model for cross-lingual entity alignment. It jointly embeds the structures of two KBs into a unified vector space and further refines it by leveraging attribute correlations in the KBs. Our experimental results on real-world datasets show that this approach significantly outperforms the state-of-the-art embedding approaches for cross-lingual entity alignment and could be complemented with methods based on machine translation.

Keywords: cross-lingual entity alignment, knowledge base embedding, joint attribute-preserving embedding

1 Introduction

In the past few years, knowledge bases (KBs) have been successfully used in lots of AI-related areas such as Semantic Web, question answering and Web mining. Various KBs cover a broad range of domains and store rich, structured real-world facts. In a KB, each fact is stated in a triple of the form $(entity, property, value)$, in which $value$ can be either a literal or an entity. The sets of entities, properties, literals and triples are denoted by E, P, L and T, respectively. Blank nodes are ignored for simplicity. There are two types of properties—relationships (R) and attributes (A)—and correspondingly two types of triples, namely relationship triples and attribute triples. A relationship triple $tr \in E \times R \times E$ describes the relationship between two entities, e.g. $(Texas, hasCapital, Austin)$, while an attribute triple $tr \in E \times A \times L$ gives a literal attribute value to an entity, e.g. $(Texas, areaTotal,$ "696241.0"$)$.

As widely noted, KBs often suffer from two problems: (i) Low coverage. Different KBs are constructed by different parties using different data sources. They contain complementary facts, which makes it imperative to integrate multiple KBs. (ii) Multi-linguality gap. To support multi-lingual applications, a growing number of multi-lingual KBs and language-specific KBs have been built. This makes it both necessary and beneficial to integrate cross-lingual KBs.

Entity alignment is the task of finding entities in two KBs that refer to the same real-world object. It plays a vital role in automatically integrating multiple KBs. This paper focuses on cross-lingual entity alignment. It can help construct a coherent KB and deal with different expressions of knowledge across diverse natural languages. Conventional cross-lingual entity alignment methods rely on machine translation, of which the accuracy is still far from perfect. Spohr *et al.* [21] argued that the quality of alignment in cross-lingual scenarios heavily depends on the quality of translations between multiple languages.

Following the popular translation-based embedding models [1,15,22], a few studies leveraged KB embeddings for entity alignment and achieved promising results [5,11]. Embedding techniques learn low-dimensional vector representations (i.e., embeddings) of entities and encode various semantics (e.g. types) into them. Focusing on KB structures, the embedding-based methods provide an alternative for cross-lingual entity alignment without considering their natural language labels.

There remain several challenges in applying embedding methods to cross-lingual entity alignment. First, to the best of our knowledge, most existing KB embedding models learn embeddings based solely on relationship triples. However, we observe that attribute triples account for a significant portion of KBs. For example, we count triples of infobox facts from English DBpedia (2016-04),[3] and find 58,181,947 attribute triples, which are three times as many as relationship triples (the number is 18,598,409). Facing the task of entity alignment, attribute triples can provide additional information to embed entities, but how to incorporate them into cross-lingual embedding models remains largely unexplored. Second, thanks to the Linking Open Data initiative, there exist some aligned entities and properties between KBs, which can serve as bridge between them. However, as discovered in [5], the existing alignment between cross-lingual KBs usually accounts for a small proportion. So how to make the best use of it is crucial for embedding cross-lingual KBs.

To deal with the above challenges, we introduce a joint attribute-preserving embedding model for cross-lingual entity alignment. It employs two modules, namely structure embedding (SE) and attribute embedding (AE), to learn embeddings based on two facets of knowledge (relationship triples and attribute triples) in two KBs, respectively. SE focuses on modeling relationship structures of two KBs and leverages existing alignment given beforehand as bridge to overlap their structures. AE captures the correlations of attributes (i.e. whether these

[3] `http://wiki.dbpedia.org/downloads-2016-04`

attributes are commonly used together to describe an entity) and clusters entities based on attribute correlations. Finally, it combines SE and AE to jointly embed all the entities in the two KBs into a unified vector space $\mathbb{R}^d$, where d denotes the dimension of the vectors. The aim of our approach is to find latent cross-lingual target entities (i.e. truly-aligned entities that we want to discover) for a source entity by searching its nearest neighbors in $\mathbb{R}^d$. We expect the embeddings of latent aligned cross-lingual entities to be close to each other.

In summary, the main contributions of this paper are as follows:

- We propose an embedding-based approach to cross-lingual entity alignment, which does not depend on machine translation between cross-lingual KBs.
- We jointly embed the relationship triples of two KBs with structure embedding and further refine the embeddings by leveraging attribute triples of KBs with attribute embedding. To the best of our knowledge, there is no prior work learning embeddings of cross-lingual KBs while preserving their attribute information.
- We evaluated our approach on real-world cross-lingual datasets from DBpedia. The experimental results show that our approach largely outperformed two state-of-the-art embedding-based methods for cross-lingual entity alignment. Moreover, it could be complemented with conventional methods based on machine translation.

The rest of this paper is organized as follows. We discuss the related work on KB embedding and cross-lingual KB alignment in Section 2. We describe our approach in detail in Section 3, and report experimental results in Section 4. Finally, we conclude this paper with future work in Section 5.

2 Related Work

We divide the related work into two subfields: KB embedding and cross-lingual KB alignment. We discuss them in the rest of this section.

2.1 KB Embedding

In recent years, significant efforts have been made towards learning embeddings of KBs. TransE [1], the pioneer of translation-based methods, interprets a relationship vector as the translation from the head entity vector to its tail entity vector. In other words, if a relationship triple (h, r, t) holds, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ is expected. TransE has shown its great capability of modeling 1-to-1 relations and achieved promising results for KB completion. To further improve TransE, later work including TransH [22] and TransR [15] was proposed. Additionally, there exist a few non-translation-based approaches to KB embedding [2,18,20].

Besides, several studies take advantage of knowledge in KBs to improve embeddings. Krompaß *et al.* [13] added type constraints to KB embedding models and enhanced their performance on link prediction. KR-EAR [14] embeds attributes additionally by modeling attribute correlations and obtains good results on predicting entities, relationships and attributes. But it only learns attribute embeddings in a single KB, which hinders its application to cross-lingual cases. Besides, KR-EAR focuses on the attributes whose values are from a small set of entries, e.g. values of "gender" are {Female, Male}. It may fail to model attributes whose values are very sparse and heterogeneous, e.g. "name", "label" and "coordinate". RDF2Vec [19] uses local information of KB structures to generate sequences of entities and employs language modeling approaches to learn entity embeddings for machine learning tasks. For cross-lingual tasks, [12] extends NTNKBC [4] for cross-lingual KB completion. [7] uses a neural network approach that translates English KBs into Chinese to expand Chinese KBs.

2.2 Cross-lingual KB Alignment

Existing work on cross-lingual KB alignment generally falls into two categories: cross-lingual ontology matching and cross-lingual entity alignment. For cross-lingual ontology matching, Fu *et al.* [8,9] presented a generic framework, which utilizes machine translation tools to translate labels to the same language and uses monolingual ontology matching methods to find mappings. Spohr *et al.* [21] leveraged translation-based label similarities and ontology structures as features for learning cross-lingual mapping functions by machine learning techniques (e.g. SVM). In all these works, machine translation is an integral component.

For cross-lingual entity alignment, MTransE [5] incorporates TransE to encode KB structures into language-specific vector spaces and designs five alignment models to learn translation between KBs in different languages with seed alignment. JE [11] utilizes TransE to embed different KBs into a unified space with the aim that each seed alignment has similar embeddings, which is extensible to the cross-lingual scenario. Wang *et al.* [23] proposed a graph model, which only leverages language-independent features (e.g. out-/inlinks) to find cross-lingual links between Wiki knowledge bases. Gentile *et al.* [10] exploited embedding-based methods for aligning entities in Web tables. Different from them, our approach jointly embeds two KBs together and leverages attribute embedding for improvement.

3 Cross-lingual Entity Alignment via KB Embedding

In this section, we first introduce notations and the general framework of our joint attribute-preserving embedding model. Then, we elaborate on the technical details of the model and discuss several key design issues.

We use lower-case bold-face letters to denote the vector representations of the corresponding terms, e.g., $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ denotes the vector representation of triple (h, r, t). We use capital bold-face letters to denote matrices, and we use superscripts to denote different KBs. For example, $\mathbf{E}^{(1)}$ denotes the representation matrix for entities in KB_1 in which each row is an entity vector $\mathbf{e}^{(1)}$.

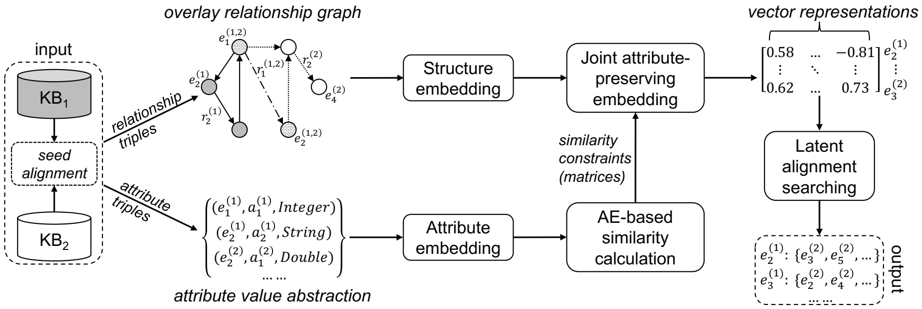
overlay relationship graph
vector representations
input
KB_1
seed alignment
KB_2
relationship triples
attribute triples
$e_1^{(1,2)}$
$e_2^{(1)}$
$r_2^{(1)}$
$r_1^{(1,2)}$
$r_2^{(2)}$
$e_4^{(2)}$
$e_2^{(1,2)}$
Structure embedding
Joint attribute-preserving embedding
$\begin{bmatrix} 0.58 & \cdots & -0.81 \\ \vdots & \ddots & \vdots \\ 0.62 & \cdots & 0.73 \end{bmatrix}$
$e_2^{(1)}$
$\vdots$
$e_3^{(2)}$
similarity constraints (matrices)
Latent alignment searching
$(e_1^{(1)}, a_1^{(1)}, Integer)$
$(e_2^{(1)}, a_2^{(1)}, String)$
$(e_2^{(2)}, a_1^{(2)}, Double)$
… …
attribute value abstraction
Attribute embedding
AE-based similarity calculation
$e_2^{(1)}$: $\{e_3^{(2)}, e_5^{(2)}, \ldots\}$
$e_3^{(1)}$: $\{e_2^{(2)}, e_4^{(2)}, \ldots\}$
… …
output

Fig. 1: Framework of the joint attribute-preserving embedding model

3.1 Overview

The framework of our joint attribute-preserving embedding model is depicted in Fig. 1. Given two KBs, denoted by KB_1 and KB_2, in different natural languages and some pre-aligned entity or property pairs (called seed alignment, denoted by superscript $^{(1,2)}$), our model learns the vector representations of KB_1 and KB_2 and expects the latent aligned entities to be embedded closely.

Following TransE [1], we interpret a relationship as the translation from the head entity to the tail entity, to characterize the structure information of KBs. We let each pair in the seed alignment share the same representation to serve as bridge between KB_1 and KB_2 to build an overlay relationship graph, and learn representations of all the entities jointly under a unified vector space via structure embedding (SE). The intuition is that two alignable KBs are likely to have a number of aligned triples, e.g. $(Washington, capitalOf, America)$ in English and its correspondence $(Washington, capitaleDes, \acute{E}tats\text{-}Unis)$ in French. Based on this, SE aims at learning approximate representations for the latent aligned triples between the two KBs.

However, SE only constrains that the learned representations must be compatible within each relationship triple, which causes the disorganized distribution of some entities due to the sparsity of their relationship triples. To alleviate this incoherent distribution, we leverage attribute triples for helping embed entities based on the observation that the latent aligned entities usually have a high degree of similarity in attribute values. Technically, we overlook specific attribute values by reason of their complexity, heterogeneity and cross-linguality. Instead, we abstract attribute values to their range types, e.g. $(Tom, age, \text{“12”})$ to $(Tom, age, Integer)$, where $Integer$ is the abstract range type of value “12”. Then, we carry out attribute embedding (AE) on abstract attribute triples to capture the correlations of cross-lingual and mono-lingual attributes, and calculate the similarities of entities based on them. Finally, the attribute similarity constraints are combined with SE to refine representations by clustering entities with high attribute correlations. In this way, our joint model preserves both relationship and attribute information of the two KBs.

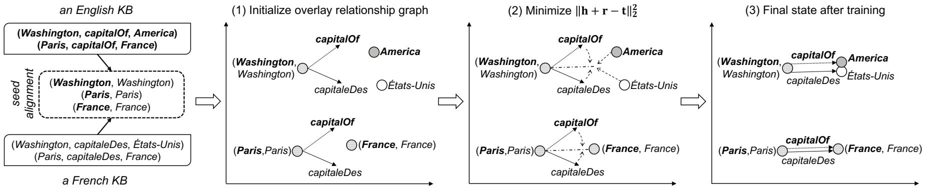

Fig. 2: An example of structure embedding

With entities represented as vectors in a unified embedding space, the alignment of latent cross-lingual target entities for a source entity can be conducted by searching the nearest cross-lingual neighbors in this space.

3.2 Structure Embedding

The aim of SE is to model the geometric structures of two KBs and learn approximate representations for latent aligned triples. Formally, given a relationship triple $tr = (h, r, t)$, we expect $\mathbf{h} + \mathbf{r} = \mathbf{t}$. To measure the plausibility of tr, we define the score function $f(tr) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2^2$. We prefer a lower value of $f(tr)$ and want to minimize it for each relationship triple.

Fig 2 gives an example about how SE models the geometric structures of two KBs with seed alignment. In Phase (1), we initialize all the vectors randomly and let each pair in seed alignment overlap to build the overlay relationship graph. In order to show the triples intuitively in the figure, we regard an entity as a point in the vector space and move relationship vectors to start from their head entities. Note that, currently, entities and relationships distribute randomly. In Phase (2), we minimize scores of triples and let vector representations compatible within each relationship triple. For example, the relationship $capitalOf$ would tend to be close to $capitaleDes$ because they share the same head entity and tail entity. In the meantime, the entity $America$ and its correspondence $\acute{E}tats$-$Unis$ would move closely to each other due to their common head entity and approximate relationships. Therefore, SE is a dynamic spreading process. The ideal state after training is shown as Phase (3). We can see that the latent aligned entities $America$ and $\acute{E}tats$-$Unis$ lie together.

Furthermore, we detect that negative triples (a.k.a. corrupted triples), which have been widely used in translation-based embedding models [1,15,22], are also valuable to SE. Considering that another English entity $China$ and its latent aligned French one $Chine$ happen to lie closely to $America$, SE may take the $Chine$ as a candidate for $America$ by mistake due to their short distance. Negative triples would help reduce the occurrence of this coincidence. If we generate a negative triple $tr' = (Washington, capitalOf, China)$ and learn a high score for tr', $China$ would keep a distance away from $America$. As we enforce the length of any embedding vector to 1, the score function f has a constant maximum. Thus, we would like to minimize $-f(tr')$ to learn a high score for tr'.

In summary, we prefer lower scores for existing triples (positives) and higher scores for negatives, which leads to minimize the following objective function:

$$\mathcal{O}_{SE} = \sum_{tr \in T} \sum_{tr' \in T'_{tr}} \big(f(tr) - \alpha f(tr')\big), \tag{1}$$

where T denotes the set of all positive triples and T'_{tr} denotes the associated negative triples for tr generated by replacing either its head or tail by a random entity (but not both at the same time). α is a ratio hyper-parameter that weights positive and negative triples and its range is $[0, 1]$. It is important to remember that each pair in the seed alignment share the same embedding during training, in order to bridge two KBs.

3.3 Attribute Embedding and Entity Similarity Calculation

Attribute Embedding We call a set of attributes correlated if they are commonly used together to describe an entity. For example, attributes *longitude*, *latitude* and *place_name* are correlated because they are widely used together to describe a place. Moreover, we want to assign a higher correlation to the pair of *longitude* and *latitude* because they have the same range type. We use seed entity pairs to establish correlations between cross-lingual attributes. Given an aligned entity pair $(e^{(1)}, e^{(2)})$, we regard the attributes of $e^{(1)}$ as correlated ones for each attribute of $e^{(2)}$, and vice versa. We expect attributes with high correlations to be embedded closely.

To capture the correlations of attributes, AE borrows the idea from Skip-gram [16], a very popular model that learns word embeddings by predicting the context of a word given the word itself. Similarly, given an attribute, AE wants to predict its correlated attributes. In order to leverage the range type information, AE minimizes the following objective function:

$$\mathcal{O}_{AE} = -\sum_{(a,c) \in H} w_{a,c} \cdot \log p(c|a), \tag{2}$$

where H denotes the set of positive (a, c) pairs, i.e., c is actually a correlated attribute of a, and the term $\log p(c|a)$ denotes the probability. To prevent all the vectors from having the same value, we adopt the negative sampling approach [17] to efficiently parameterize Eq. (2), and $\log p(c|a)$ is replaced with the term as follows:

$$\log \sigma(\mathbf{a} \cdot \mathbf{c}) + \sum_{(a,c') \in H'_a} \log \sigma(-\mathbf{a} \cdot \mathbf{c}'), \tag{3}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$. H'_a is the set of negative pairs for attribute a generated according to a log-uniform base distribution, assuming that they are all incorrect.

We set $w_{a,c} = 1$ if a and c have different range types, otherwise $w_{a,c} = 2$ to increase their probability of tending to be similar. In this paper, we distinguish four kinds of abstract range types, i.e., *Integer*, *Double*, *Datetime* and *String* (as default). Note that it is easy to extend to more types.

Entity Similarity Calculation Given attribute embeddings, we take the representation of an entity to be the normalized average of its attribute vectors, i.e., $\mathbf{e} = \left[\sum_{a \in A_e} \mathbf{a}\right]_1$, where A_e is the set of attributes of e and $[.]_1$ denotes the normalized vector. We have two matrices of vector representations for entities in two KBs, $\mathbf{E}_{AE}^{(1)} \in \mathbb{R}^{n_e^{(1)} \times d}$ for KB_1 and $\mathbf{E}_{AE}^{(2)} \in \mathbb{R}^{n_e^{(2)} \times d}$ for KB_2, where each row is an entity vector, and $n_e^{(1)}$, $n_e^{(2)}$ are the numbers of entities in KB_1, KB_2, respectively.

We use the cosine distance to measure the similarities between entities. For two entities e, e', we have $\text{sim}(e, e') = \cos(\mathbf{e}, \mathbf{e}') = \frac{\mathbf{e} \cdot \mathbf{e}'}{||\mathbf{e}||||\mathbf{e}'||} = \mathbf{e} \cdot \mathbf{e}'$, as the length of any embedding vector is enforced to 1. The cross-KB similarity matrix $\mathbf{S}^{(1,2)} \in \mathbb{R}^{n_e^{(1)} \times n_e^{(2)}}$ between KB_1 and KB_2, as well as the inner similarity matrices $\mathbf{S}^{(1)} \in \mathbb{R}^{n_e^{(1)} \times n_e^{(1)}}$ for KB_1 and $\mathbf{S}^{(2)} \in \mathbb{R}^{n_e^{(2)} \times n_e^{(2)}}$ for KB_2, are defined as follows:

$$\mathbf{S}^{(1,2)} = \mathbf{E}_{AE}^{(1)} \mathbf{E}_{AE}^{(2)\top}, \quad \mathbf{S}^{(1)} = \mathbf{E}_{AE}^{(1)} \mathbf{E}_{AE}^{(1)\top}, \quad \mathbf{S}^{(2)} = \mathbf{E}_{AE}^{(2)} \mathbf{E}_{AE}^{(2)\top}. \tag{4}$$

A similarity matrix $\mathbf{S}$ holds the cosine similarities among entities and $\mathbf{S}_{i,j}$ is the similarity between the i-th entity in one KB and the j-th entity in the same or the other KB. We discard lower values of $\mathbf{S}$ because a low similarity of two entities indicates that they are likely to be different. So, we set the entry $\mathbf{S}_{i,j} = 0$ if $\mathbf{S}_{i,j} < \tau$, where τ is a threshold and can be set based on the average similarity of seed entity pairs. In this paper, we fix $\tau = 0.95$ for inner similarity matrices and 0.9 for cross-KB similarity matrix, to achieve high accuracy.

3.4 Joint Attribute-Preserving Embedding

We want similar entities across KBs to be clustered to refine their vector representations. Inspired by [25], we use the matrices of pairwise similarities between entities as supervised information and minimize the following objective function:

$$\begin{aligned} \mathcal{O}_S = & \left\|\mathbf{E}_{SE}^{(1)} - \mathbf{S}^{(1,2)} \mathbf{E}_{SE}^{(2)}\right\|_F^2 \\ & + \beta\left(\left\|\mathbf{E}_{SE}^{(1)} - \mathbf{S}^{(1)} \mathbf{E}_{SE}^{(1)}\right\|_F^2 + \left\|\mathbf{E}_{SE}^{(2)} - \mathbf{S}^{(2)} \mathbf{E}_{SE}^{(2)}\right\|_F^2\right), \end{aligned} \tag{5}$$

where β is a hyper-parameter that balances similarities between KBs and their inner similarities. $\mathbf{E}_{SE} \in \mathbb{R}^{n_e \times d}$ denotes the matrix of entity vectors for one KB in SE with each row an entity vector. $\mathbf{S}^{(1,2)} \mathbf{E}_{SE}^{(2)}$ calculates latent vectors of entities in KB_1 by accumulating vectors of entities in KB_2 based on their similarities. By minimizing $\left\|\mathbf{E}_{SE}^{(1)} - \mathbf{S}^{(1,2)} \mathbf{E}_{SE}^{(2)}\right\|_F^2$, we expect similar entities across KBs to be embedded closely. The two inner similarity matrices work in the same way.

To preserve both the structure and attribute information of two KBs, we jointly minimize the following objective function:

$$\mathcal{O}_{joint} = \mathcal{O}_{SE} + \delta \mathcal{O}_S, \tag{6}$$

where δ is a hyper-parameter weighting $\mathcal{O}_S$.

3.5 Discussions

We discuss and analyze our joint attribute-preserving embedding model in the following aspects:

Objective Function for Structure Embedding SE is translation-based embedding model but its objective function (see Eq. (1)) does not follow the margin-based ranking loss function below, which is used by many previous KB embedding models [1]:

$$\mathcal{O} = \sum_{tr \in T} \sum_{tr' \in T'_{tr}} \max[\gamma + f(tr) - f(tr'), 0]. \tag{7}$$

Eq. (7) aims at distinguishing positive and negative triples, and expects that their scores can be separated by a large margin. However, for the cross-lingual entity alignment task, in addition to the large margin between their scores, we also want to assign lower scores to positive triples and higher scores to negative triples. Therefore, we choose Eq. (1) instead of Eq. (7).

In contrast, JE [11] uses the margin-based ranking loss from TransE [1], while MTransE [5] does not have this as it does not use negative triples. However, as explained in Section 3.2, we argue that negative triples are effective in distinguishing the relations between entities. Our experimental results reported in Section 4.4 also demonstrate the effectiveness of negative triples.

Training We initialize parameters such as vectors of entities, relations and attributes randomly based on a truncated normal distribution, and then optimize Eqs. (2) and (6) with a gradient descent optimization algorithm called AdaGrad [6]. Instead of directly optimizing $\mathcal{O}_{joint}$, our training process involves two optimizers to minimize $\mathcal{O}_{SE}$ and $\delta\mathcal{O}_S$ independently. At each epoch, the two optimizers are executed alternately. When minimizing $\mathcal{O}_{SE}$, $f(tr)$ and $-\alpha f(tr')$ can also be optimized alternately.

The length of any embedding vector is enforced to 1 for the following reasons: (i) this constraint prevents the training process from trivially minimizing the objective function by increasing the embedding norms and shaping the embeddings, (ii) it limits the randomness of entity and relationship distribution in the training process, and (iii) it fixes the mismatch between the inner product in Eq. (3) and the cosine similarity to measure embeddings [24].

Our model is also scalable in training. The structure embedding belongs to the translation-based embedding models, which have already been proved to be capable of learning embeddings at large scale [1]. We use sparse representations for matrices in Eq. (5) for saving memory. Additionally, the memory cost to compute Eq. (4) can be reduced using a divide-and-conquer strategy.

Parameter Complexity The parameter complexity of our joint model is $O\big(d(n_e + n_r + n_a)\big)$, where n_e, n_r, n_a are the numbers of entities, relationships

Table 1: Statistics of the datasets

Datasets		Entities	Relationships	Attributes	Rel. triples	Attr. triples
DBP15K$_{\text{ZH-EN}}$	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
DBP15K$_{\text{JA-EN}}$	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
DBP15K$_{\text{FR-EN}}$	French	66,858	1,379	4,547	192,191	528,665
	English	105,889	2,209	6,422	278,590	576,543

and attributes, respectively. d is the dimension of the embeddings. Considering that $n_r, n_a \ll n_e$ in practice and the seed alignment share vectors in training, the complexity of the model is roughly linear to the number of total entities.

Searching Latent Aligned Entities Because the length of each vector always equals 1, the cosine distance between entities of the two KBs can be calculated as $\mathbf{D} = \mathbf{E}_{SE}^{(1)}\mathbf{E}_{SE}^{(2)\top}$. Thus, the nearest entities can be obtained by simply sorting each row of $\mathbf{D}$ in descending order. For each source entity, we expect the rank of its truly-aligned target entity to be the first few.

4 Evaluation

In this section, we report our experiments and results on real-world cross-lingual datasets. We developed our approach, called JAPE, using TensorFlow[4]—a very popular open-source software library for numerical computation. Our experiments were conducted on a personal workstation with an Intel Xeon E3 3.3 GHz CPU and 128 GB memory. The datasets, source code and experimental results are accessible at this website[5].

4.1 Datasets

We selected DBpedia (2016-04) to build three cross-lingual datasets. DBpedia is a large-scale multi-lingual KB including inter-language links (ILLs) from entities of English version to those in other languages. In our experiments, we extracted 15 thousand ILLs with popular entities from English to Chinese, Japanese and French respectively, and considered them as our reference alignment (i.e., gold standards). Our strategy to extract datasets is that we randomly selected an ILL pair s.t. the involved entities have at least 4 relationship triples and then extracted relationship and attribute infobox triples for selected entities. The statistics of the three datasets are listed in Table 1, which indicate that the number of involved entities in each language is much larger than 15 thousand, and attribute triples contribute to a significant portion of the datasets.

[4] `https://www.tensorflow.org/`

[5] `https://github.com/nju-websoft/JAPE`

4.2 Comparative Approaches

As aforementioned, JE [11] and MTransE [5] are two representative embedding-based methods for entity alignment. In our experiments, we used our best effort to implement the two models as they do not release any source code or software currently. We conducted them on the above datasets as comparative approaches. Specifically, MTransE has 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. We followed the implementation details reported in [5,11] and complemented other unreported details with careful consideration. For example, we added a strong orthogonality constraint for the linear transformation matrix in MTransE to ensure the invertibility, because we found it leads to better results. For JAPE, we tuned various parameter values and set $d = 75, \alpha = 0.1, \beta = 0.05, \delta = 0.05$ for the best performance. The learning rates of SE and AE were empirically set to 0.01 and 0.1, respectively.

4.3 Evaluation Metrics

Following the conventions [1,5,11], we used $Hits@k$ and $Mean$ to assess the performance of the three approaches. $Hits@k$ measures the proportion of correctly aligned entities ranked in the top k, while $Mean$ calculates the mean of these ranks. A higher $Hits@k$ and a lower $Mean$ indicate better performance. It is a phenomenon worth noting that the optimal $Hits@k$ and $Mean$ usually do not come at the same epoch in all the three approaches. For fair comparison, we did not fix the number of epochs but used early stopping to avoid overtraining. The training process is stopped as long as the change ratio of $Mean$ is less than 0.0005. Besides, the training of AE on each dataset takes 100 epochs.

4.4 Experimental Results

Results on DBP15K We used a certain proportion of the gold standards as seed alignment while left the remaining as testing data, i.e., the latent aligned entities to discover. We tested the proportion from 10% to 50% with step 10%, and Table 2 lists the results using 30% of the gold standards. The variation of $Hits@k$ with different proportions will be shown shortly. For relationships and attributes, we simply extracted the property pairs with exactly the same labels, which only account for a small portion of the seed alignment.

Table 2 indicates that JAPE largely outperformed JE and MTransE, since it captures both structure and attribute information of KBs. For JE, it employs TransE as its basic model, which is not suitable to be directly applied to entity alignment as discussed in Section 3.5. Besides, JE does not give a mandatory constraint on the length of vectors. Instead, it only minimizes $\|\mathbf{v}\|_2^2 - 1$ to restrain vector length and brings adverse effect. For MTransE, it models the structures of KBs in different vector spaces, and information loss happens when learning the translation between vector spaces.

Table 2: Result comparison and ablation study

$\text{DBP15K}_{\text{ZH-EN}}$		ZH → EN				EN → ZH			
		Hits@1	*Hits*@10	*Hits*@50	*Mean*	*Hits*@1	*Hits*@10	*Hits*@50	*Mean*
JE		21.27	42.77	56.74	766	19.52	39.36	53.25	841
MTransE		30.83	61.41	79.12	154	24.78	52.42	70.45	208
JAPE	SE w/o neg.	38.34	68.86	84.07	103	31.66	59.37	76.33	147
	SE	39.78	72.35	87.12	84	32.29	62.79	80.55	109
	SE + AE	**41.18**	**74.46**	**88.90**	**64**	**40.15**	**71.05**	**86.18**	**73**

$\text{DBP15K}_{\text{JA-EN}}$		JA → EN				EN → JA			
		Hits@1	*Hits*@10	*Hits*@50	*Mean*	*Hits*@1	*Hits*@10	*Hits*@50	*Mean*
JE		18.92	39.97	54.24	832	17.80	38.44	52.48	864
MTransE		27.86	57.45	75.94	159	23.72	49.92	67.93	220
JAPE	SE w/o neg.	33.10	63.90	80.80	114	29.71	56.28	73.84	156
	SE	34.27	66.39	83.61	104	31.40	60.80	78.51	127
	SE + AE	**36.25**	**68.50**	**85.35**	**99**	**38.37**	**67.27**	**82.65**	**113**

$\text{DBP15K}_{\text{FR-EN}}$		FR → EN				EN → FR			
		Hits@1	*Hits*@10	*Hits*@50	*Mean*	*Hits*@1	*Hits*@10	*Hits*@50	*Mean*
JE		15.38	38.84	56.50	574	14.61	37.25	54.01	628
MTransE		24.41	55.55	74.41	139	21.26	50.60	69.93	156
JAPE	SE w/o neg.	29.55	62.18	79.36	123	25.40	56.55	74.96	133
	SE	29.63	64.55	81.90	95	26.55	60.30	78.71	107
	SE + AE	**32.39**	**66.68**	**83.19**	**92**	**32.97**	**65.91**	**82.38**	**97**

Additionally, we divided JAPE into three variants for ablation study, and the results are shown in Table 2 as well. We found that involving negative triples in structure embedding reduces the random distribution of entities, and involving attribute embedding as constraint further refines the distribution of entities. The two improvements demonstrate that systematic distribution of entities makes for the cross-lingual entity alignment task.

It is worth noting that the alignment direction (e.g. ZH → EN vs. EN → ZH) also causes performance difference. As shown in Table 1, the relationship triples in a non-English KB are much sparser than those in an English KB, so that the approaches based on the relationship triples cannot learn good representations to model the structures of non-English KBs, as restraints for entities are relatively insufficient. When performing alignment from an English KB to a non-English KB, we search for the nearest non-English entity as the aligned one to an English entity, the sparsity of the non-English KB leads to the disorganized distribution of its entities, which brings negative effects on the task. However, it is comforting to see that the performance difference becomes narrower when involving attribute embedding, because the attribute triples provide additional information to embed entities, especially for sparse KBs.

Fig. 3 provides the visualization of sample results for entity alignment and attribute correlations. We projected the embeddings of aligned entity pairs and involved attribute embeddings to two dimensions using PCA. The left part indicates that universities, countries, cities and cellphones were divided widely while aligned entities from Chinese to English were laid closely, which met our expectation of JAPE. The right part shows our attribute embedding clustered three

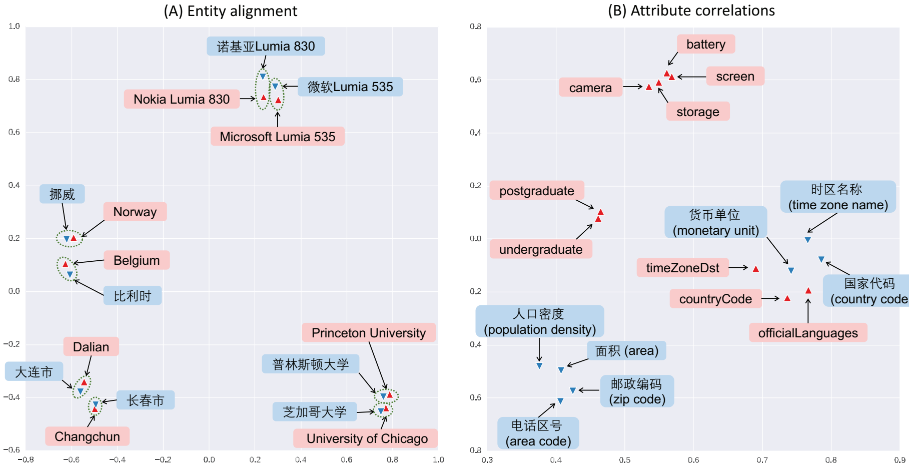

Fig. 3: Visualization of results on DBP15K$_{\text{ZH-EN}}$

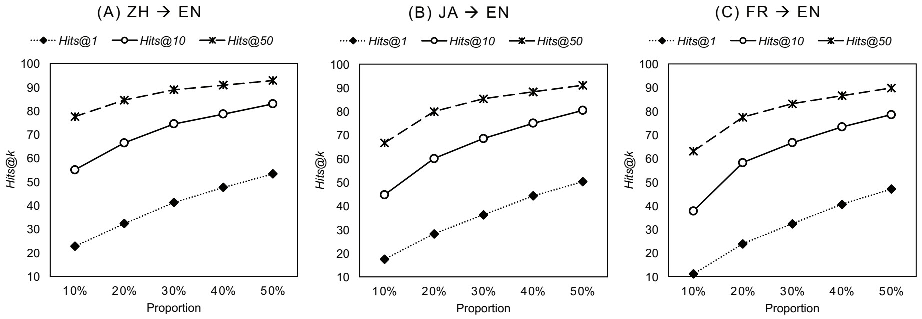

Fig. 4: $Hits@k$ w.r.t. proportion of seed alignment

groups of monolingual attributes (about cellphones, cities and universities) and one group of cross-lingual ones (about countries).

Sensitivity to Proportion of Seed Alignment Fig. 4 illustrates the change of $Hits@k$ with varied proportion of seed alignment. In accordance with our expectation, the results on all the datasets become better with the increase of the proportion, because more seed alignment can provide more information to overlay the two KBs. It can be seen that, when using half of the gold standards as seed alignment, JAPE performed encouragingly, e.g. $Hits@1$ and $Hits@10$ on DBP15K$_{\text{ZH-EN}}$ are 53.27% and 82.91%, respectively. Moreover, even with a very small proportion of seed alignment like 10%, JAPE still achieved promising results, e.g. $Hits@10$ on DBP15K$_{\text{ZH-EN}}$ reaches 55.04% and on DBP15K$_{\text{JA-EN}}$ reaches 44.69%. Therefore, it is feasible to deploy JAPE to various entity alignment tasks, even with limited seed alignment.

Table 3: Combination of machine translation and JAPE

DBP15K$_{\text{ZH-EN}}$	ZH → EN				EN → ZH			
	Hits@1	*Hits*@10	*Hits*@50	*Mean*	*Hits*@1	*Hits*@10	*Hits*@50	*Mean*
Machine translation	55.76	67.61	74.30	820	40.38	54.27	62.27	1,551
JAPE	41.18	74.46	88.90	64	40.15	71.05	86.18	73
Combination	**73.09**	**90.43**	**96.61**	**11**	**62.70**	**85.21**	**94.25**	**26**

DBP15K$_{\text{JA-EN}}$	JA → EN				EN → JA			
	Hits@1	*Hits*@10	*Hits*@50	*Mean*	*Hits*@1	*Hits*@10	*Hits*@50	*Mean*
Machine translation	74.64	84.57	89.13	333	61.98	72.07	77.22	1,095
JAPE	36.25	68.50	85.35	99	38.37	67.27	82.65	113
Combination	**82.84**	**94.65**	**98.31**	**9**	**75.94**	**90.70**	**96.04**	**25**

Combination with Machine Translation Since machine translation is often used in cross-lingual ontology matching [9,21], we designed a machine translation based approach that employs Google Translate to translate the labels of entities in one KB and computes similarities between the translations and the labels of entities in the other KB. For similarity measurement, we chose Levenshtein distance because of its popularity in ontology matching [3].

We chose DBP15K$_{\text{ZH-EN}}$ and DBP15K$_{\text{JA-EN}}$, which have big barriers in linguistics. As depicted in Table 3, machine translation achieves satisfying results, especially for *Hits*@1, and we think that it is due to the high accuracy of Google Translate. However, the gap between machine translation and JAPE becomes smaller for *Hits*@10 and *Hits*@50. The reason is as follows. When Google misunderstands the meaning of labels (e.g. polysemy), the top-ranked entities are all very likely to be wrong. On the contrary, JAPE relies on the structure information of KBs, so the correct entities often appear slightly behind. Besides, we found that translating from Chinese (or Japanese) to English is more accurate than the reverse direction.

To further investigate the possibility of combination, for each latent aligned entities, we considered the lower rank of the two results as the combined rank. It is surprising to find that the combined results are significantly better, which reveals the mutual complementarity between JAPE and machine translation. We believe that, when aligning entities between cross-lingual KBs where the quality of machine translation is difficult to guarantee, or many entities lack meaningful labels, JAPE can be a practical alternative.

Results at Larger Scale To test the scalability of JAPE, we built three larger datasets by choosing 100 thousand ILLs between English and Chinese, Japanese and French in the same way as DBP15K. The threshold of relationship triples to select ILLs was set to 2. Each dataset contains several hundred thousand entities and several million triples. We set $d = 100, \beta = 0.1$ and keep other parameters the same as DBP15K. For JE, the training takes 2000 epochs as reported in its paper. The results on DBP100K are listed in Table 4. Due to lack of space, only *Hits*@10 is reported. We found that similar results and conclusions stand for

Table 4: *Hits*@10 comparison on DBP100K

DBP100K	ZH → EN	EN → ZH	JA → EN	EN → JA	FR → EN	EN → FR
JE	16.95	16.63	21.17	20.98	22.98	22.63
MTransE	34.31	29.18	33.93	27.22	44.84	39.19
JAPE	**41.75**	**40.13**	**42.00**	**39.30**	**53.64**	**50.51**

DBP100K compared with DBP15K, which indicate the scalability and stability of JAPE.

Furthermore, the performance of all the methods decreases to some extent on DBP100K. We think that the reasons are twofold: (i) DBP100K contains quite a few "sparse" entities involved in a very limited number of triples, which affect embedding the structure information of KBs; and (ii) as the number of latent aligned entities in DBP100K are several times larger than DBP15K, the TransE-based models suffer from the increased occurrence of multi-mapping relations as explained in [22]. Nevertheless, JAPE still outperformed JE and MTransE.

5 Conclusion and Future Work

In this paper, we introduced a joint attribute-preserving embedding model for cross-lingual entity alignment. We proposed structure embedding and attribute embedding to represent the relationship structures and attribute correlations of KBs and learn approximate embeddings for latent aligned entities. Our experiments on real-world datasets demonstrated that our approach achieved superior results than two state-of-the-art embedding approaches and could be complemented with conventional methods based on machine translation.

In future work, we look forward to improving our approach in several aspects. First, the structure embedding suffered from multi-mapping relations, thus we plan to extend it with cross-lingual hyperplane projection. Second, our attribute embedding discarded attribute values due to their diversity and cross-linguality, which we want to use cross-lingual word embedding techniques to incorporate. Third, we would like to evaluate our approach on more heterogeneous KBs developed by different parties, such as between DBpedia and Wikidata.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Nos. 61370019, 61572247 and 61321491).

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS. pp. 2787–2795 (2013)
2. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: AAAI. pp. 301–306 (2011)
3. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: Alani, H., et al. (eds.) ISWC. pp. 294–309 (2013)

4. Chen, D., Socher, R., Manning, C.D., Ng, A.Y.: Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. arXiv:1301.3618 (2013)
5. Chen, M., Tian, Y., Yang, M., Zaniolo, C.: Multi-lingual knowledge graph embeddings for cross-lingual knowledge alignment. In: IJCAI (2017)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12(7), 2121–2159 (2011)
7. Feng, X., Tang, D., Qin, B., Liu, T.: English-chinese knowledge base translation with neural network. In: COLING. pp. 2935–2944 (2016)
8. Fu, B., Brennan, R., O'Sullivan, D.: Cross-lingual ontology mapping – an investigation of the impact of machine translation. In: Gómez-Pérez, A., et al. (eds.) ASWC. pp. 1–15 (2009)
9. Fu, B., Brennan, R., O'Sullivan, D.: Cross-lingual ontology mapping and its use on the multilingual semantic web. In: WWW Workshop on Multilingual Semantic Web. pp. 13–20 (2010)
10. Gentile, A.L., Ristoski, P., Eckel, S., Ritze, D., Paulheim, H.: Entity matching on web tables : a table embeddings approach for blocking. In: EDBT. pp. 510–513 (2017)
11. Hao, Y., Zhang, Y., He, S., Liu, K., Zhao, J.: A joint embedding method for entity alignment of knowledge bases. In: Chen, H., et al. (eds.) CCKS. pp. 3–14 (2016)
12. Klein, P., Ponzetto, S.P., Glavaš, G.: Improving neural knowledge base completion with cross-lingual projections. In: EACL. pp. 516–522 (2017)
13. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., et al. (eds.) ISWC. pp. 640–655 (2015)
14. Lin, Y., Liu, Z., Sun, M.: Knowledge representation learning with entities, attributes and relations. In: IJCAI. pp. 2866–2872 (2016)
15. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. pp. 2181–2187 (2015)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS. pp. 3111–3119 (2013)
18. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: ICML. pp. 809–816 (2011)
19. Ristoski, P., Paulheim, H.: Rdf2vec: RDF graph embeddings for data mining. In: Groth, P., et al. (eds.) ISWC. pp. 498–514 (2016)
20. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: NIPS. pp. 926–934 (2013)
21. Spohr, D., Hollink, L., Cimiano, P.: A machine learning approach to multilingual and cross-lingual ontology matching. In: Aroyo, L., et al. (eds.) ISWC. pp. 665–680 (2011)
22. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. pp. 1112–1119 (2014)
23. Wang, Z., Li, J., Wang, Z., Tang, J.: Cross-lingual knowledge linking across wiki knowledge bases. In: WWW. pp. 459–468 (2012)
24. Xing, C., Wang, D., Liu, C., Lin, Y.: Normalized word embedding and orthogonal transform for bilingual word translation. In: HLT-NAACL. pp. 1006–1011 (2015)
25. Zou, W.Y., Socher, R., Cer, D.M., Manning, C.D.: Bilingual word embeddings for phrase-based machine translation. In: EMNLP. pp. 1393–1398 (2013)