COTSAE: CO-Training of Structure and Attribute Embeddings for Entity Alignment

Kai Yang[1,2,4], **Shaoqin Liu**[1,2,4], **Junfeng Zhao**[1,2,4*], **Yasha Wang**[1,2,3], **Bing Xie**[1,2]

[1] Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China
[2] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[3] National Engineering Research Center For Software Engineering, Peking University, 100871
[4] Peking University Information Technology Institute (Tianjin Binhai), Tianjin 300450, China
{smileyk, liu_shaoqin, zhaojf, wangyasha, xiebing} @pku.edu.cn

Abstract

Entity alignment is a fundamental and vital task in Knowledge Graph (KG) construction and fusion. Previous works mainly focus on capturing the structural semantics of entities by learning the entity embeddings on the relational triples and pre-aligned "seed entities". Some works also seek to incorporate the attribute information to assist refining the entity embeddings. However, there are still many problems not considered, which dramatically limits the utilization of attribute information in the entity alignment. Different KGs may have lots of different attribute types, and even the same attribute may have diverse data structures and value granularities. Most importantly, attributes may have various "contributions" to the entity alignment. To solve these problems, we propose COTSAE that combines the structure and attribute information of entities by co-training two embedding learning components, respectively. We also propose a *joint attention* method in our model to learn the attentions of attribute types and values cooperatively. We verified our COTSAE on several datasets from real-world KGs, and the results showed that it is significantly better than the latest entity alignment methods. The structure and attribute information can complement each other and both contribute to performance improvement.

Introduction

In the past few decades, more and more knowledge graphs (KGs) have been built in the domain of finance, entertainment, government, science, and biomedicine, as they could play essential roles in information retrieval, recommendation system, machine comprehension, and knowledge-based QA-system. Different institutions construct their KGs according to different needs and data sources, that leads to the diverse forms of KGs, even in the same domain. However, these KGs also have many repetitions and can complement each other. Before merging different KGs into a more unified one, the primary task is to identify the entities in different KGs that refer to the same real-world object, which is called entity alignment. For example, the entity "www.dbpedia.org/resource/The-Three-Troubledoers" in Dbpedia KG and the entity

"www.wikidata.org/entity/Q7769108" in Wikidata KG both refer to the American movie "The Three Troubledoers." Entity alignment is to find these entity pairs between KGs by learning the semantics of entities.

According to (Lacostejulien et al. 2013), the semantic of entity is implicated by two kinds of information. One is the relations between the entities, which is also called structure information; the other is the attribute information describing the self-characteristics of the entities. Two kinds of information are independent of each other in one KG, but describe the internal and external semantics of entities, respectively. Due to the different application scenarios, some KGs may focus on the construction of entity relations, and some KGs pay more attention to the integrity of entity attributes.

Currently, the works of entity alignment mainly focus on the utilization of structural information. They try to learn the entity embeddings to represent the structural semantics on the relational triples $\langle h, r, t \rangle$ in KGs. For example, JAPE(Hao et al. 2016) uses TransE(Bordes et al. 2013) to interpret a relationship r as a transition from its head entity h to its tail entity t, $\vec{h}+\vec{r} \approx \vec{t}$. They use the pre-aligned "seed alignments" as the "bridge" of embedding spaces by keeping the embeddings of the "seed entities" same in different KGs. Then entities with similar embeddings could be considered as the same entities to be aligned. Since such methods do not depend on the content of the KGs, they are mostly used to solve the cross-lingual entity alignment. However, TransE model 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 (Sun, Wei, and Li 2017). Moreover, it is difficult for the seed entities to guarantee the coverage of all relation types, which also severely limits the accuracy of such works.

The attribute information can give an alternative view of entities especially in the mono-lingual entity alignment. Recently, using attribute embeddings to align entities concentrates on the attribute types (Sun, Wei, and Li 2017; Hao et al. 2017; Xu et al. 2019). For example, in attribute triple (Tom,age,12), the "age" is an attribute type of the entity "Tom", and "12" is the value of "age". AttrE(Trsedya, Qi, and Zhang) needs to match the attribute types before utiliz-

---

ing attribute values, which is not practical for the attributes with different naming conventions in many large-scale KGs. Meanwhile, the attributes of an entity is treated equally in these, which leads to bad performance of entity alignment by learning a large number of noise attributes. In short, there are still many problems in the use of entity attribute information for embedding-based entity alignment:

- The attribute information is pretty challenging to use as different KGs may have lots of different attribute types, and the name of the same attribute type may also be different due to naming conventions. Attribute values are diverse in data structure and value granularity.

- Different attributes contributes differently to the entity alignment, and there are often a large number of "noise" attributes that are unrelated to the entity alignment but interfering with the alignment.

- Attribute and structure information describes the internal and external semantics of an entity, and they are independent of each other in the KGs. Actually, there are different emphasis on these two information in different KGs. The structure information of some KGs may be sparse, or some KGs may be lack of attribute information. If learning them into one unified embedding space by constraining the embeddings from different information consistent, e.g., AttrE (Trsedya, Qi, and Zhang), sparse structure information will limit the learning of structure embedding, which definitely affects the learning of attribute embedding.

To solve the above challenges, we propose the COTSAE, which is a Co-Training model including two embedding components to integrate structure and attribute information in this paper. One component is the KG representation learning model TransE (Bordes et al. 2013) to learn the structural embedding of entities. The other component is a novel Pseudo-Siamese network with joint attention to learn the weights of attributes and bi-directional gated recurrent unit encoder (Bi-GRU) to characterize attribute values. In COTSAE, both structure and attribute information can be used to generate possible entity alignment, for example, when the structure information of KGs is difficult to get new entity alignments, the attribute information can be used to complement it for further entity alignment. Similarly, when the attribute information is insufficient, the structure information can also assist it in the entity alignment. The main contributions of our approach are list as followed:

- We propose a novel Pseudo-Siamese network to make full use of the attribute types and values in KGs. To learn the importance of attributes for entity alignment, we propose a joint attention method in which the attribute type and its corresponding value learn one attention weight jointly. We also use the Bi-GRUs to capture the forward and backward sequence information of attribute values.

- We propose COTSAE, a Co-Training framework learning entity embeddings using the TransE component and Pseudo-Siamese network alternately to integrate the structure and attribute information. In each iteration, it selects entity pairs with similar embeddings as the new "seed alignments", and supplements the structure and attribute information of the entities in training dataset. These two informations can complement and reinforce each other in the iterative entity alignment.

- We validated COTSAE on large-scale and small-scale datasets constructed from real-world KGs, respectively. Experiments show that COTSAE is superior to other entity alignment methods apparently. The results of entity alignment based only on structure information or attribution information indicates that the two types of information can effectively enhance each other.

The rest of the paper is organized as follows. We first discuss the related works and then introduce our approach in the section that follows. After that, we present the experimental results and conclude the paper in the last section.

Related Work

KG Embedding In the past few years, knowledge graph embedding models have been widely used in the KG completion task. It indicates that the embedding model can effectively use the structural information in the knowledge graph to predict the missing entities or relationships. The translation model proposed by TransE (Bordes et al. 2013) regards the relationship in the triple (h, r, t) as a translation between the head entity h and the tail entity t: $\vec{h} + \vec{r} \approx \vec{t}$. TransE is straightforward and effective, but it is difficult to model complex relationships (e.g., one-to-many relations). Researchers have proposed many improved models upon TransE, such as TransH(Zhen et al. 2014), TransR(Lin et al. 2015) and TransD(Ji et al. 2015), etc. Compared with the TransE, these models embed entity and relation into different embedding spaces which increases the ability of the model to express complex relationships. In addition to the translation model, there are also many non-translation models(Socher et al. 2013; Dettmers et al. 2017).

Embedding-based entity alignment The embedding-based methods for entity alignment are generally divided into the embedding learning phase and the entity alignment phase. The embedding learning phase uses knowledge representation learning methods such as the TransE to learn entity embeddings for KGs. The entity alignment phase obtains matched entity pairs by calculating the similarity of each two entity embeddings. MtransE(Chen et al. 2016) uses the TransE model to learn the embeddings of entities in the two KGs and then learns a transformation function connecting two embedding spaces for the entity alignment. Some works use semi-supervised learning to overcome the sparsity of "seed alignments" for the entity alignment. In IPTransE(Hao et al. 2017) and BootEA(Sun et al. 2018), the newly discovered entity alignments in the iterative process are added to the training dataset to optimize the following process. BootEA(Sun et al. 2018) also adds an error correction mechanism to avoid the impact of error accumulation. Some works attempt to improve the entity representation learning model to make it more suitable for the entity alignment. (Liang et al. 2018; Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017) utilize the graph neural
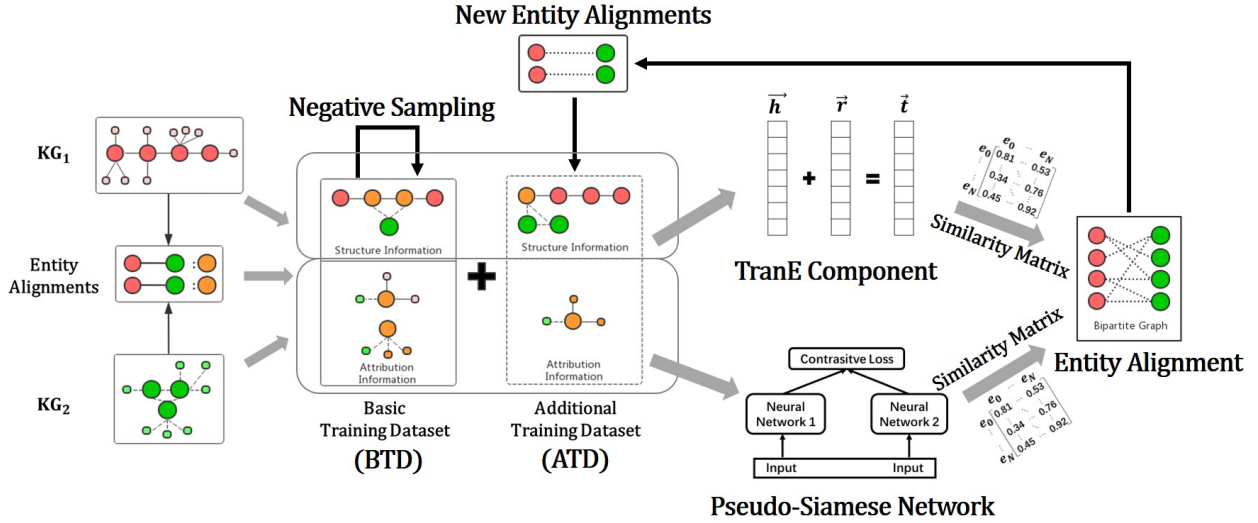
Figure 1: Brief Framework of the COTSAE

network (e.g., GCN) to propagate the profiles/labels/degrees information of nodes, and learn the node embeddings for n-ode classification. To better obtain the information of neighbor entities and consider the multi-hop path, GCN-Align(Xu et al. 2019) uses the graph convolution network to obtain the embeddings of the entities. RSN4EA(Guo, Sun, and Hu 2019) uses the sampled multi-hop paths as the training dataset to learn the representations of the entities. These works mainly rely on relationship triples, which may cause the disorganized distribution of some entities due to the sparsity of their relationship triples. Many related works incorporate other information in the KG with structural information to improve the performance of the entity alignment. JAPE(Sun, Wei, and Li 2017) learns the embeddings of attributes to capture the correlations of attributes using the Skip-Gram method. Then it combines the structure embedding and attribute embedding to find similar entities. KDCoE(Chen et al. 2018) utilizes the entity descriptions as the supplementary data to refine the entity embeddings. It used Co-training method to iteratively train two embedding models on multi-lingual KG structures and entity descriptions respectively. In AttrE(Trsedya, Qi, and Zhang), the attribute triples (h, a, v) is also treated like the relational triples and learn the entity embeddings by the objective function $\vec{h} + \vec{a} \approx \vec{v}$. The attribute value v here is processed as a sequence of characters. MuitiKE (Zhang et al. 2019) learns the representations of the entities in the three views: name, attribute and structure. The combination strategies are proposed to integrate three view-specific embeddings to get the latest performance of the entity alignment. The attribute is crucial information, especially in the mono-lingual entity alignment. Most of the previous works focus on the attribute type-level information of the entities, which is blurred for the entity alignment. The AttrE(Trsedya, Qi, and Zhang) method considers the utilization of attribute values, but it needs to match the attribute types in advance, which is not

practical for the attributes with different naming conventions in many large-scale KGs. MultiKE(Zhang et al. 2019) uses externally trained word embeddings to refine the learning of attribute values. But it still treats all attributes of the entity equally, which inevitably limit the performance of the entity alignment due to the large percentage of "noise" attributes.

Proposed Approach

We start with the definition of the entity alignment , and then we present an overview of our proposed model COTSAE. We detail the components of the proposed model afterward, including the training dataset generation, the TransE component and the Pseudo-Siamese network. Finally, we introduce the entity alignment process briefly.

Problem Definition

Let E_1 be the entity set of KG_1 and E_2 be the entity set of KG_2. The aim of entity alignment is to find every entity pairs $A = \{(e_1, e_2)|e_1 \; queals \; to \; e_2, e_1 \in E_1, e_2 \in E_2\}$. In general, a subset pre-aligned entity pairs $\mathcal{P}$ of A is used as "seed alignments" for training data. The rest of A denoting the unaligned entities are what we aim to align. We model entity alignment as a classification problem of using entities in e_2 to label entities in e_1(Sun et al. 2018). By convention, we consider the so-called one-to-one entity alignment: an entity can be associated with at most one label(Sun et al. 2018; Lacostejulien et al. 2013; Zhang et al. 2015).

Approach Overview

The brief framework of our COTSAE is depicted in Figure 1. **Firstly**, given two KGs (KG_1, KG_2) with structure and attribute information and "seed alignments" $\mathcal{P}$, we construct the Basic Training Dataset (BTD) for two model components respectively. **Then** in each iteration, the entity embeddings are learned either on the structure information of the

BTD using the TransE component or on attribute information of the BTD using the Pseudo-Siamese neural network. **After that**, all candidate alignments are generated in the form of the bipartite graph by computing the embedding similarities of entity pairs and filtering them with a threshold γ. **Finally** the new entity alignments of this iteration can be obtained to form the Additional Training Dataset (ATD) which is the complements of the BTD using the best bipartite graph matching algorithm. In the COTSAE model, utilizing the structure and attribute information alternately to align the entities can make these two informations complement and reinforce each other by extending the training dataset iteratively.

Iterative Co-training & Training Data Generation

As is shown in Figure 1, training dataset includes BTD and ATD; both of them contain structure and attribute information for TransE component and Pseudo-Siamese Network component, respectively. The co-training of the two components is conducted iteratively on the KGs, where a small amount of "seed alignments "$\mathcal{P}$ is also provided. The entities involved in "seed alignment" are the anchors help find potential alignments. At each iteration, the component models alternately take turns of the train-and-predict process. In train process, the TransE component or the Pseudo-Siamese Network is trained for several epochs with early-stopping on the BDT and ADT. The ADT is empty at the first iteration. After training, the model predicts new entity alignments for ADT to supplement the BDT. Both components conduct the above train-and-predict process alternately, therefore gradually reinforce the entity alignment, until two components no longer predict new entity alignments.

Following (Sun et al. 2018), the new relational triples are generated by exchanging the "seed entities" with its pre-aligned entities in relational triples. Given the "seed alignments" $\mathcal{P} = \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2, e_1 \text{ equales to } e_2\}$, The new relational triples are $\mathbb{S}^{\mathcal{P}} = \bigcup_{(e_1,e_2) \in \mathcal{P}} \mathbb{S}_{(e_1,e_2)}$, where $\mathbb{S}_{(e_1,e_2)}$ is:

$$\{(e_2,r,t)|(e_1,r,t) \in \mathbb{S}_1^+\} \cup \{(h,r,e_2)|(h,r,e_1) \in \mathbb{S}_1^+\} \cup$$
$$\{(e_1,r,t)|(e_2,r,t) \in \mathbb{S}_2^+\} \cup \{(h,r,e_1)|(h,r,e_2) \in \mathbb{S}_2^+\}.$$

$\mathbb{S}_1^+$ and $\mathbb{S}_2^+$ denote the relational triples of KG_1 and KG_2. Therefore the positive samples of structure information $\mathbb{S}^+ = \mathbb{S}_1^+ \cup \mathbb{S}_2^+ \cup \mathbb{S}^{\mathcal{P}}$. We also use the ϵ-Truncated Uniform Negative Sampling method following (Bordes et al. 2013) to generate the negative relational triples $\mathbb{S}^-$ for the training dataset. We sample the negative triples from the top-n nearest neighbors in the relational embedding space, where n is decided by the ϵ.

Same as structural triples, the attribute triples is: $\mathbb{A}^+ = \bigcup_{(e_1,e_2) \in \mathcal{P}} \mathbb{A}_{(e_1,e_2)}$, where $\mathbb{A}_{(e_1,e_2)}$ is $\{(e_1,a,v)|(e_1,a,v) \in \mathbb{A}_1^+\} \cup \{(e_2,a,v)|(e_2,a,v) \in \mathbb{A}_2^+\}$. But the input data of our Pseudo-Siamese Neural Network is the attribute types and values separately instead of attribute triples. The attribute triples describe more detailed and accurate information about entities, which is harder to learn than the relational triples. Thus we use the nearest
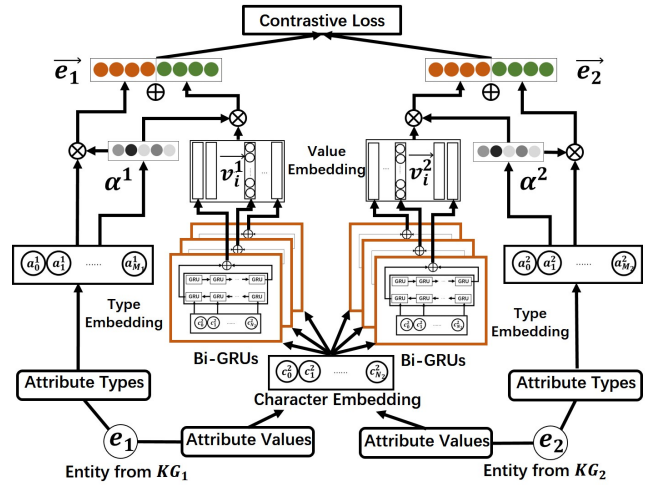

Figure 2: Proposed Pseudo-Siamese Network Architecture for Attribute Information

neighbor as the exchange of each "seed entity" to generate the negative samples $\mathbb{A}^-$ instead of sampling from a limited scope. To lighten the "semantic drift" when extending the BTD with ATD, we do not generate the negative samples for both structure and attribute information in ATD.

TransE Component

The intuition of using structural information to align entities is that two alignable entities are likely to have similar relations with other entities. Based on this, TransE is employed to learn structural embeddings for the entity alignments. To reduce the "semantic drift" of embeddings in each iteration and better capture the semantics of entities in the unified space, we use two absolute thresholds $\gamma_1, \gamma_2 > 0$ as the margins to for the scores function $d(h+r,t) = \|\vec{h}+\vec{r}-\vec{t}\|_2^2$. According to (Sun et al. 2018), we minimize the two-margin-based objective function:

$$\mathcal{L}_S = \sum_{(h,r,t) \in \mathbb{S}^+} \sum_{(h',r',t') \in \mathbb{S}^-} [d(h+r,t) - \gamma_1]_+ \quad (1)$$
$$+ \mu_1 [\gamma_2 - d(h'+r',t')]_+$$

where $[.] = max(0,.)$ and μ_1 is a hyper-parameter to control the weight of negative samples.

In each iteration of COTSAE, the input of TransE component is the structure information from BTD and ATD. TransE component learns a low-dimensional embedding to describe the structural information of entities in KGs.

Pseudo-Siamese Neural Network

We leverage attribute information based on the observation that the to-be-aligned entities usually have high similarity in their attributes. However, the attribute information is challenging to use due to its complexity, heterogeneity, and a large percent of noise. We proposed a novel method based on Pseudo-Siamese Neural Network with two identical but not same modeling streams, which considers both attribute type and attribute values.

After first been proposed in (Bromley et al. 1993), the Siamese Neural Networks have been used successfully for various tasks, such as person re-identification (Yi et al. 2014), speaker verification(Ke and Salman 2011), and face identification(Yi, Wang, and Tang 2014). The Siamese Network often consists of two sub-networks with shared parameters that process each data patch independently. The Pseudo-Siamese Network is essentially a Siamese Network, but without sharing parameters, i.e., In COTSAE, we set different sub-networks includding the attribte type embeddings and value embeddings to handle attribute types and values with different lengths and granularities. The sub-networks for the two KGs only share the parameters of the character embedding.

In our Pseudo-Siamese network, the entity embeddings are learned on the attribute information of two KGs and then evaluates the similarity between the entity embeddings using some trained metrics. As is depicted in Figure 2, the Bi-GRUs are used to capture the character-level information of attribute values. We calculate the attention of attributes with attribute type and share the attention weights with the corresponding attribute values, which is called the *joint attention* method. We compute the Euclidean distance of entities from KG_1 and KG_2, $d(\vec{e}_1, \vec{e}_2) = \|\vec{e}_1 - \vec{e}_2\|_2$. And the loss function of our Pseudo-Siamese Neural Network is the contrastive loss :

$$\mathcal{L}_a = \frac{1}{2K} \sum_{i=0}^{K} y^{(i)} d(e_1^{(i)}, e_2^{(i)})^2 + (1 - y^{(i)}) max(\gamma_3 \\ - d(e_1^{(i)}, e_2^{(i)}))^2 \tag{2}$$

where K is the size of training dataset. $(e_1^{(i)}, e_2^{(i)}; y^{(i)})$ is i-th training sample, in which $e_1^{(i)}$ is the entity of KG_1, $e_2^{(i)}$ is the entity of KG_2 and $y^{(i)}$ is the alignment label of this entity pair. γ_3 is the margin of the contrastive loss.

Character-Level Attribute Value Embedding   The same attribute value may appear in different forms in two KGs, e.g., 50.9989 vs. 509988 as the area of a city entity; "Barack Obama" vs. "Barack Hussein Obama" as the name of a person entity, etc. In mono-lingual KGs, the attribute value can be seen as a sequence of characters with the same vocabulary. Therefore we use bi-directional Gated Recurrent Unit (Bi-GRU) network to encode the attribute value from two direction into the single embeddings.

The GRU cell is defined by the following equations:

$$z_i = \sigma(\mathbf{W}_z[c_i, h_{i-1}]) \tag{3}$$

$$r_i = \sigma(\mathbf{W}_r[c_i, h_{i-1}]) \tag{4}$$

$$\widetilde{h}_i = tanh(\mathbf{W}_h[c_i, r_i \odot h_{i-1}]) \tag{5}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \widetilde{h}_i \tag{6}$$

where z and r are the update gate and the reset gate of the GRU cell.$\mathbf{W}_z$, $\mathbf{W}_r$ and $\mathbf{W}_h$ are weight matrices.

The Bi-GRU consists of a forward GRU and a backward GRU. The forward GRU reads the input character embeddings, e.g., the attribute value $\vec{v} = (\vec{c_0}, \vec{c_2}, ..., \vec{c_n})$, from left to right and the backward GRU reads character embeddings

reversely. The output of forward GRU and backward GRU is:

$$\vec{h}_f = \mathbf{GRU}(c_n, \vec{h}_{f-1}) \; ; \; \vec{h}_b = \mathbf{GRU}(c_0, \vec{h}_{b+1}) \tag{7}$$

The initial states of the Bi-GRU are set to zero vectors. After reading the character embeddings, we concatenate the final hidden states of two-directional GRU networks to form the attribute value embedding, $\vec{v} = \left[\vec{h}_f; \vec{h}_b\right]$. Then the entities e_1 and e_2 can get their attribute value embeddings $\vec{\mathcal{V}}^1 = (\vec{v}_0^1, \vec{v}_1^1, ..., \vec{v}_{M_1}^1)$ and $\vec{\mathcal{V}}^2 = (\vec{v}_0^2, \vec{v}_1^2, ..., \vec{v}_{M_2}^2)$.

Joint Attention for Attribute types and Values   There is a large percent of "noise" attributes in an entity that contribute little to the entity alignment. For example, entity "Titanic" in DBPedia has an attribution "IMDB_ID" to record its ID in the IMDB website, but the "IMDB_ID" attribute does not exist for "Titanic" in YAGO. Here "IMDB_ID" in DbPedia can be seen as a noise attribute which is not essential for aligning the entity "Titanic". To learn the importance of different attributes for an entity, we proposed a joint attention method, which makes the attribute type and attribute value shared one attention weight.

Given attribute type embedding $\mathcal{A} = (\vec{a}_0, \vec{a}_1, ..., \vec{a}_M)$ of entity e with M attributes, we calculate the attention weight α_i of i-th attribute:

$$\alpha_i = softmax(\mathcal{A}^T \mathbf{W}_a \vec{a}_i) \tag{8}$$

where $\mathbf{W}_a$ denotes the weight matrix of $\vec{a}_i$. Here we use the attribute type embedding to get the attention weight, and the weight of attribute value should be consistent with that of its attribute type:

$$e_{type} = \sum_{i=0}^{M} \alpha_i \vec{a}_i \; ; \; e_{value} = \sum_{i=0}^{M} \alpha_i \vec{v}_i \tag{9}$$

where $\vec{a}_i$ is the i-th attribute type embedding and $\vec{v}_i$ is its responding attribute value embedding.

Therefore we can get the final entity embedding by concatenating the attribute type embedding and attribute value embedding: $\vec{e}_1 = [e_{type}^1 || e_{value}^1]$ and $\vec{e}_2 = [e_{type}^2 || e_{value}^2]$.

Entity Alignment

We can learn the entity's embedding by either the TransE component or the pseudo-siamese network, and obtain the similarity matrics of every entity pairs. **Firstly**, we use the threshold γ_4 to filter entity pairs and construct a bipartite graph of entity alignment by adding an edge to each potential pair of entities. **Then**, the best bipartite graph matching algorithm is used to find the global optimum result of this iteration as the new entity alignments for the next iteration. It is hard to ensure that the additional training set is completely correct. When it contains a large amount of error information, it will affect the following training of our model, leading to the error accumulation called "semantic drift" of the entity alignment. **Therefore**, when the result of the entity alignment in different iterations conflicts(Sun et al. 2018), the additional training set is continuously modified according to the similarity of the entity pairs.

Table 1: Statistics of the entity alignment datasets

Datasets		#Ent.	#Rel.	#Attr.	#Rel tr.	#Attr tr.
DWY15K						
DBP-WD-N	DBpedia	15,000	253	343	38,421	69,641
	Wikidata	15,000	144	613	40,159	128,737
DBP-WD-D	DBpedia	15,000	220	236	68,598	70,548
	Wikidata	15,000	135	435	75,465	125,752
DBP-YG-N	DBpedia	15,000	219	324	33,571	66,279
	YAGO3	15,000	30	34	34,660	56,319
DBP-YG-D	DBpedia	15,000	206	246	71,257	78,990
	YAGO3	15,000	30	34	97,131	58,412
DWY100K						
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

Experiments

In this section, we report our experiments and results compared with several state-of-art methods on a set of real-world KG datasets. We used Tensorflow to develop our approach COTSAE[1]. Our experiments were conducted on a server with an Intel Xeon E5-2620 2.1 GHz CPU, 4 NVIDIA GeForce Titan xp GPU and 128 GB memory.

Datasets

To verify the performance of our model on the KGs with different data scales and entity sampling distributions, we use two datasets:

- DWY15K contains four couples of datasets with different entity sampling distribution extracted from DBpedia, Wikidata and YAGO3, denoted by DBP-WD-Norm, DBP-WD-Dense, DBP-YG-Norm, and DBP-YG-Dense. Each dataset has 15 thousand reference entity alignment. The entities and structural triples are from (Guo, Sun, and Hu 2019) which guarantee the degree distributions of the sampled entities following the original KGs. We then extract all the attribute triples that involve the entities in the alignments from the original KGs(DBpedia, Wikidata, and YAGO3).

- DWY100K contains two large-scale datasets which are also extracted from DBpedia, Wikidata and YAGO3. Each dataset has 100 thousand reference entity alignment. The entities and structural triples are from (Sun et al. 2018). Similarly, we extract all the attribute triples that involve the entities in the alignments from the original KGs(DBpedia, Wikidata, and YAGO3).

The statistical data of DWY15K and DWY100K is shown in Table 1. Each dataset provides 30% entities as "seed alignments" by default and leaves the remaining for evaluating entity alignment performance.

Experiment Setting

For comparison, we chose several state-of-art embedding-based approaches to entity alignment: MTransE (Chen et

---

[1]https://github.com/ykpku/COTSA

Table 2: Entity Alignment Results on DWY100K

Approaches	DP-WD			DP-YG		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
MTransE	28.12	51.59	0.363	25.15	49.29	0.334
IPTransE	34.85	63.84	0.447	29.74	55.76	0.386
BootEA	74.79	89.84	0.801	76.10	89.44	0.808
GCN-Align	47.70	75.96	0.577	60.05	84.14	0.686
JAPE	31.84	58.88	0.411	23.57	48.41	0.320
AttrE	38.96	66.77	0.487	23.24	42.70	0.300
MulKE	91.45	95.19	0.928	88.03	95.32	0.906
COTSAE_a	81.75	92.71	0.818	78.29	91.86	0.831
COTSAE_sa(NGram)	81.35	93.89	0.857	89.40	96.46	0.919
COTSAE_sa(BiGRU)	84.09	95.16	0.876	85.88	95.21	0.892
COTSAE_san	**92.68**	**97.86**	**0.945**	**94.39**	**98.74**	**0.961**

al. 2016), IPTransE (Hao et al. 2017), JAPE (Sun, Wei, and Li 2017), BootEA (Sun et al. 2018), GCN-Align (Xu et al. 2019), AttrE (Trsedya, Qi, and Zhang), and MultiKE(Zhang et al. 2019), which have been discussed in the Related Work section. As the textual descriptions of sampled entities are needed, which is not available, we skipped KDCoE(Chen et al. 2018). In order to compare(Zhang et al. 2019) considering the name, attribute, and relations of entities, we also add the name information by just computing the edit distance of entities at the beginning of our approach. We tried our best to tune the hyper-parameters for all the methods.

The hyper-parameters of COTSAE were used as below. For the TransE component model, we followed (Sun et al. 2018) and set $\gamma_1 = 0.01$, $\gamma_2 = 2.0$, and $\mu_1 = 0.2$. And 10 negative relational triples were sampled for each positive triple. The entity dimensions of entities and relations were set to 75. The learning rate is set to 0.01, and the batch size is 2000. In Pseudo-Siamese Neural Network, we choose the most 51 frequent used characters, and the character embedding size is set to 32. The embedding size of attribute type is set to 64. The learning rate of Pseudo-Siamese Neural Network is set as 0.001, and the batch is set as 64. We set the weight size of joint attention as 32 and the size of the hidden layer of Bi-GRUs as 64.

Due to the different convergence rate of TransE component and Pseudo-Siamese Neural Network, in each iteration of COTSAE, we first train the TransE component for 10 epochs and generate new negative samples every 5 epoches according to the entity embeddings. Then we train the Pseudo-Siamese Neural Network for 20 epochs and generate its negative sample every 10 epoches.

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.

For an ablation study, We built a set of variants of COTSAE: COTSAE_a uses only the attribute information in the Pseudo-Siamese Neural Network for entity alignment and expend training set iteratively. COTSAE_sa is a complete model of the COTSA method, containing structural and attribute information with Co-Training method. In AttrE (Trsedya, Qi, and Zhang), N-gram-based compositional function achieves better performance than using the LSTM-based function while mapping attribute values sequence

Table 3: Entity Alignment Results on DWY15K

Approaches	DP-WD-N			DP-YG-N			DP-WD-D			DP-YG-D		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
MTransE	22.3	50.1	0.32	24.6	54.0	0.34	38.9	68.7	0.49	22.8	51.3	0.32
IPTransE	23.1	51.7	0.33	22.7	50.0	0.32	43.5	74.5	0.54	23.6	51.3	0.33
BootEA	32.3	63.1	0.42	31.3	62.5	0.42	67.8	91.2	0.76	68.2	89.8	0.76
GCN-Align	17.7	37.8	0.25	19.3	41.5	0.27	43.1	71.3	0.53	31.3	57.5	0.40
RSNs	38.8	65.7	0.49	40.0	67.5	0.50	76.3	92.4	0.83	82.6	95.8	0.87
JAPE	21.9	50.1	0.31	23.3	52.7	0.33	39.3	70.5	0.50	26.8	57.3	0.37
AttrE	11.9	30.5	0.18	17.3	33.5	0.23	43.3	72.5	0.56	26.7	46.1	0.33
COTSAE_a	35.7	38.9	0.37	36.4	47.0	0.41	60.8	90.9	0.61	63.5	71.6	0.67
COTSAE_sa(NGram)	41.3	69.9	0.51	53.4	77.1	0.61	74.7	91.2	0.81	90.3	97.4	0.93
COTSAE_sa(BiGRU)	42.3	70.3	0.51	44.7	70.3	0.53	82.3	95.4	0.87	87.3	96.6	0.91
COTSAE_san	**70.9**	**90.4**	**0.77**	**73.8**	**93.9**	**0.81**	**92.2**	**98.3**	**0.94**	**97.6**	**98.9**	**0.98**
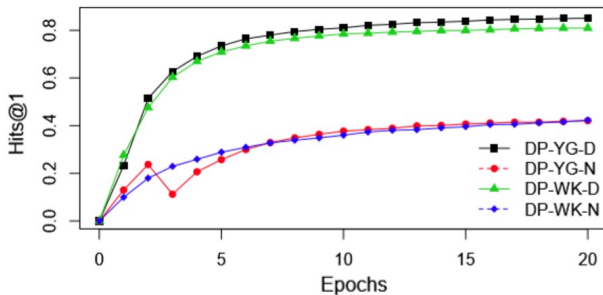
Figure 3: Hits@1 results w.r.t. epochs required by alternative components to converge

Figure 4: Hits@1 results w.r.t. proportion of "seed alignments"

to their vector representations. Thus we build the COT-SAE_sa(NGram) with N-gram-based compositional function to compare with the COTSAE_sa(BiGRU) model. To compare with MultiKE (Zhang et al. 2019), we also establish COTSAE_san, which adds the name information of the entity based on COTSAE_sa. Detailly, we first used the edit distance to calculate the similarity of entity names. When the entity pairs with similarity higher than 0.98 are used as the pre-aligned entity, then conduct the COTSAE model to align the remaining entities.

Entity Alignment Results

Tables 2 and Table 3 show the alignment results of the proposed COTSAE models comparing to other embedding based methods on the DWY15K and DWY100K datasets. We found that the performances of COTSAEs are significantly better than the state-of-art methods. In the comparisons, MTransE, IPTransE, BootEA, GCN-Align, and RSNs only use the structural information of the entities in the KG. Our COTSAE_sa is better than these methods because we take advantage of the attribute information of entities, which gives more help for accurate alignment when the structural information is insufficient. Except for the structural information, JAPE utilizes the type-level attributes, which is a kind of fuzzy attribute information, and in some cases may even have an adverse effect. AttrE combines the structure and attribute information in mono-lingual entity alignments.
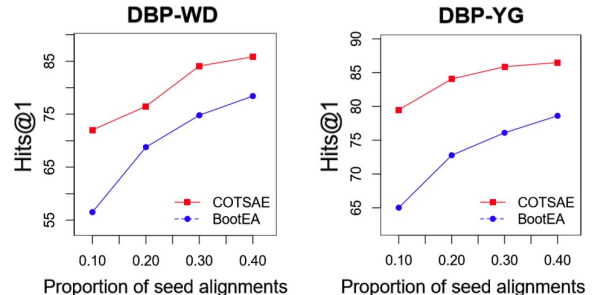
However, it requires pre-aligning the attribute types using the edit distance, which does not work in many complicated large-scale KGs. At the same time, we can see that Multi-KE, which uses three kinds of information such as name, attribute, and entity structure, has achieved latest results. The entity name is very accurate information in DBpedia, Wiki-Data, and Yago data which is also verified in MultiKE's experiments. When our COTSAE_san considers the entity name before alignments using simple edit distance, better performance of entity alignment is achieved, which means that our COTSAE_sa can effectively utilize the structure and attribute information in KGs. The results obtained by merely using the attribute information (e.g., COTSAE_a) are better than using the structural information(e.g., BootEA), indicating that in the mono-lingual entity alignment, the attribute information is more important than the structural information. From Tables 2 and Table 3, we also find that on DP-WK datasets the performance of COTSAE_sa with Bi-GRU function is better and on DP-YG datasets the performance of COTSAR_sa with N-Gram is better. In Table 2, the accuracy of the entity alignment is relatively low on dense KGs, and our COTSAE_sa is still more than 4 percentage points better than other methods with only structure and attribute information.

Discussion

In Figure 3, The accuracy of each training iteration on the test dataset is demonstrated. When the two components of COTSAE do not converge in the early stage of training, the alignment results did not appear to fluctuate violently, indicating that the COTSAE effectively lighten the "semantic drift" problem. And when the entity alignment tends to be stable after training on the structural information, the attribute information gives a new "growth momentum" to it. And the structural information also promotes the results of entity alignment on the attribute information. It shows that the structural and attribute information can reinforce each other in the entity alignment process. In the above experiments, 30% entities in the KGs were used as seeds. However, the acquisition of aligned entities is costly. We also set up experiments that reduce the proportion of seed entities in the dataset and observe the decline of the alignment results. Figure 4 shows that our COTSAE still achieves the best performance when the proportion of aligned seeds is reduced.

Conclusion

In this paper, we proposed COTSAE, a Co-Training model that integrates structure and attribute information for entity alignment in KGs. The attribute embedding component of COTSAE is based on the pseudo-siamese neural network to encode character-level attribute information. Moreover, we also proposed the joint attention method to ensure that attribute type shares the attention weight with its corresponding attribute value. Our proposed model outperforms the baselines consistently, especially in terms of hits@1 for the alignment of entities on real-world knowledge graphs.

Acknowledgments

References

Bordes, A.; Usunier, N.; Garcadurn, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *International Conference on Neural Information Processing Systems*.

Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; Lecun, Y.; Moore, C.; Sckinger, E.; and Shah, R. 1993. Signature verification using a siamese time delay neural network. In *International Conference on Neural Information Processing Systems*.

Chen, M.; Tian, Y.; Yang, M.; and Zaniolo, C. 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment.

Chen, M.; Tian, Y.; Chang, K.-W.; Skiena, S.; and Zaniolo, C. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2017. Convolutional 2d knowledge graph embeddings.

Guo, L.; Sun, Z.; and Hu, W. 2019. Learning to exploit long-term relational dependencies in knowledge graphs.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

Hao, Y.; Zhang, Y.; He, S.; Kang, L.; and Zhao, J. 2016. A joint embedding method for entity alignment of knowledge bases. In *China Conference on Knowledge Graph & Semantic Computing*.

Hao, Z.; Xie, R.; Liu, Z.; and Sun, M. 2017. Iterative entity alignment via joint knowledge embeddings. In *International Joint Conference on Artificial Intelligence*.

Ji, G.; He, S.; Xu, L.; Kang, L.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Meeting of the Association for Computational Linguistics the International Joint Conference on Natural Language Processing*.

Ke, C., and Salman, A. 2011. Extracting speaker-specific information with a regularized siamese deep network. In *International Conference on Neural Information Processing Systems*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lacostejulien, S.; Palla, K.; Davies, A.; Kasneci, G.; Graepel, T.; and Ghahramani, Z. 2013. Sigma: Simple greedy matching for aligning large knowledge bases. In *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*.

Liang, J.; Jacobs, P.; Sun, J.; and Parthasarathy, S. 2018. Semi-supervised embedding in attributed networks with outliers. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 153–161. SIAM.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth Aaai Conference on Artificial Intelligence*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. Y. 2013. Reasoning with neural tensor networks for knowledge base completion. In *International Conference on Neural Information Processing Systems*.

Sun, Z.; Hu, W.; Zhang, Q.; and Qu, Y. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*.

Sun, Z.; Wei, H.; and Li, C. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding.

Trsedya, B. D.; Qi, J.; and Zhang, R. Entity alignment between knowledge graphs using attribute embeddings.

Xu, K.; Wang, L.; Yu, M.; Feng, Y.; Song, Y.; Wang, Z.; and Yu, D. 2019. Cross-lingual knowledge graph alignment via graph matching neural network.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Deep metric learning for person re-identification. In *International Conference on Pattern Recognition*.

Yi, S.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification.

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

Zhang, Q.; Sun, Z.; Hu, W.; Chen, M.; Guo, L.; and Qu, Y. 2019. Multi-view knowledge graph embedding for entity alignment. *CoRR* abs/1906.02390.

Zhen, W.; Zhang, J.; Feng, J.; and Zheng, C. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-eighth Aaai Conference on Artificial Intelligence*.