Research Article

CAREA: Cotraining Attribute and Relation Embeddings for Cross-Lingual Entity Alignment in Knowledge Graphs

Baiyang Chen [ID],[1] **Xiaoliang Chen** [ID],[1] **Peng Lu** [ID],[2] **and Yajun Du** [ID][1]

[1]*School of Computer and Software Engineering, Xihua University, Chengdu 610039, China*
[2]*Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C3J7, Canada*

Correspondence should be addressed to Xiaoliang Chen; xdxlchen@gmail.com

Knowledge graphs (KGs) are one of the most widely used techniques of knowledge organizations and have been extensively used in many application fields related to artificial intelligence, for example, web search and recommendations. Entity alignment provides a useful tool for how to integrate multilingual KGs automatically. However, most of the existing studies evaluated ignore the abundant information of entity attributes except for entity relationships. This paper sets out to investigate cross-lingual entity alignment and proposes an iterative cotraining approach (CAREA) to train a pair of independent models. The two models can extract the attribute and the relation features of multilingual KGs, respectively. In each iteration, the two models alternate to predict a new set of potentially aligned entity pairs. Besides, this method further filters through the dynamic threshold value to enhance the two models' supervision. Experimental results on three real-world datasets demonstrate the effectiveness and superiority of the proposed method. The CAREA model improves the performance with at least an absolute increase of 3.9% across all experiment datasets. The code is available at https://github.com/ChenBaiyang/CAREA.

1. Introduction

Knowledge graphs (KGs) that possess machine-readable representations of factual knowledge are becoming the basis for many applications such as web search (Google and Bing), recommendations (Amazon and eBay), and social networks (Facebook and Linkedin). Multilingual KGs (e.g., DBpedia [1], YAGO [2], and ConceptNet [3]) are constructed in separate languages from various data sources and contain a wealth of complementary facts. The bridging of language gaps and the improvement of user experience from downstream cross-language applications benefit a lot from the entity equivalent in multilingual KGs. Hence, aligning the entities in multilingual KGs has recently attracted an increasing amount of research attention and is called the problem of cross-lingual entity alignment.

Most existing entity alignment methods entirely rely on the graph structures, while the abundant attribute information in KGs remains unexplored. The attributes of an entity represented by different languages often share enormous semantic information, leading to a potentially valid view of the entities connected to multilanguage KGs. However, it is nontrivial to capture and make use of such information for cross-lingual entity alignment. First, attribute information can be quite diverse across different KGs. The most likely cause of the differences is that there exist distinct attribute concerns in the process of developing applications. Second, the semantic association of attributes cannot be modeled directly since the critical entity expression languages are different. Moreover, the simultaneous use of relationships and attributes across multilingual KGs is a near term challenge in the area of knowledge graphs.

Cotraining is a popular machine learning method, where two complementary models utilize a large number of unlabeled examples to bootstrap the performance of each other iteratively [4, 5]. Cotraining can be readily applied to multilingual tasks since the data in these tasks have two or more views (i.e., a subset of features). It is also applicable to employ cotraining to the task of entity alignment across

multilingual KGs, as the entity attributes and graph structure information naturally form two independent views of a KG. In the cotraining framework, each model is trained on one of the two views, under the assumption that either view is sufficient to make a prediction. In each iteration, the cotraining algorithm selects high-confidence samples ranked by each of the models to form new auto-labeled data samples and then uses both labeled data and additional auto-labeled data to update the other model.

This paper introduces a cotraining based approach CAREA to learn embeddings from two independent views of knowledge (relationships and attributes) in multilingual KGs. CAREA iteratively trains two-component models that are called attribute-based model f_{attr} and structure-based model f_{struc}, respectively. f_{attr} extracts the attribute features according to attribute occurrence frequencies and value data types, which also employs a Multilayer Perceptron (MLP) to transform both KGs into a unified vector space. On the other hand, f_{struc} adopts a graph attention mechanism to capture the multirelation characteristics of KGs. During each iteration of the cotraining process, both models alternately predict a set of new potential aligned entity pairs to strengthen the supervision of cross-lingual learning. Such collaborative predictions gradually improve the performance of each model. To improve the accuracy of aligned entity prediction, we further evaluate the predicted entity pairs through a dynamic threshold. Experimental results on three real-world datasets demonstrate the effectiveness and superiority of our proposed method CAREA.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 formally defines the research question. Section 4 introduces the proposed approach. Section 5 presents the experimental results. Finally, we conclude this work in Section 6.

2. Related Work

2.1. KG Embedding. Embedding-based entity analysis approaches have demonstrated their effectiveness in modeling the semantic information of KGs, which aim to project entities into low-dimensional embedding spaces. The KG embedding model TransE [6] interprets a relation as the translation from an entity to another. Such KG embedding models using the translations have shown their feasibility and later been improved by several subsequent studies, such as TransH [7], TransR [8], and TransD [9].

TransH and TransR update the modeling of multi-mapping relations of TransE from one to many. TransD uses a dynamic matrix to transfer entities and relations rather than a fixed one. R-GCN [10] is a similar model that incorporates relation type information by setting a transformation matrix for each relation. Some authors consider avoiding the use of translation approaches for KG embedding, including [11–15]. The perfect example is shown in the study of Nathani et al. [15], which extended the graph attention mechanism (GAT) to capture entity and relationship features in the multihop neighbors of a given entity. Some research below makes use of the additional information in KGs to improve embedding performance. For example,

reverse triples and relational paths are combined in PTransE [16]. The categorical attributes such as gender and hobby are introduced in KR-EAR [17]. In addition, some works explore the type, local structure, and global patterns in KG embedding [10, 18–20].

2.2. Entity Alignment. Entity alignment aims to automatically determine whether an entity pair in different KGs refers to the same entity in reality. Traditional entity alignment methods take advantage of various features of KGs, such as the semantics of OWL properties [21], compatible neighbors and attribute values of entities [22], and the relation structures [23].

Many recent studies have used embedding methods to deal with the alignment problem in KGs. MTransE [24] deploys three mechanisms, distance-based axis alignment, translation vectors, and linear transformations to learn multilanguage KG embeddings. An improved model IPTransE [25] combines the advantages of TransE and PTransE to embed a single KG. Then, an iterative and parameter sharing step is added in IPTransE for various KGs embedding. BootEA [26] improves on JAPE [27] by using the bootstrapping strategy, which provides an iterative data labeling method. Accordingly, the constructed training data for potential entity alignment can be used to learn entity alignment-oriented embedding. MuGNN [28] learns alignment-oriented KG embeddings by a multichannel mechanism that encodes KGs via KG completion and entity pruning. NAEA [29] merges neighborhood subgraph-level information of entities and designs a neighborhood-aware attention representation mechanism on cross-lingual KGs. RDGCN [30] proposes a relation-aware dual-graph convolutional network to leverage relations through attentive interactions between the KG and its dual relation counterpart. MRAEA [31] learns cross-lingual entity embeddings by attending over the entity's neighbors and the meta semantics of its connecting relations.

Some literature on cross-lingual entity alignments has highlighted the role of both KG structures and attributes. JAPE [27] embeds the structures of different KGs into a uniform hidden space and uses the attribute correlation of KGs to realize the refinement of entity embedding. However, attribute components can significantly degrade the performance of JAPE's structural components when attributes are heterogeneous or have a confused association between the attributes. Graph convolutional networks (GCNs) [32] are also employed in the study [33] to learn embeddings from both the structure and attribute information of entities for cross-lingual alignment.

3. Problem Definition

In a KG, facts are mainly stored in two types of triples < entity, attribute, value > and < entity, relation, entity >, which are called attribute triple and relation triple, respectively. This paper denotes a KG as $G = (E, R, A)$, where $E = \{e_1, e_2, \ldots, e_N\}$ is the set of entities, $R = \{(e_i, e_j) | e_i t, ne_j q \in hE\}$ is the set of relations,

and $A = \{a_{e_i} | e_i t \in nE\}$ represents the set of attributes in the KG. Each attribute of an entity consists of a set of key-value pairs.

Definition 1. Cross-lingual entity alignment: let $G^{(1)} = (E^{(1)}, R^{(1)}, A^{(1)})$ and $G^{(2)} = (E^{(2)}, R^{(2)}, A^{(2)})$ be two arbitrary KGs in different languages. The entity pairs that refer to the same real-world object are called prealigned entities, denoted as $L = \{(e_i^{(1)}, e_j^{(2)}) | e_i^{(1)} t \in nE^{(1)}q, he_j^{(2)} \in xE^{(2)}\}$. The task of cross-lingual entity alignment is to find hidden aligned entity pairs $M = \{(e_i^{(1)}, e_j^{(2)}) | e_i^{(1)} t \in nE^{(1)}q, he_j^{(2)} \in xE^{(2)}7, C(e_i^{(1)}, e_j^{(2)}); \notin L\}$ based on the prealigned pairs L.

4. Proposed Approach

4.1. Overview. The details of the proposed model CAREA taken in this section are based on the cotraining algorithm. Its framework is shown in Figure 1.

We construct two independent models: attribute-based model f_{attr} and relation-based model f_{struc}. The advantage of the cotraining algorithm is that it reinforces the performance of the two models in the process of iterations. Both models are retrained with the prealigned entity pairs L and predict the new pairs of potential aligned entities at each iteration. Subsequently, dynamic thresholds are used to filter the anticipated results further. The method merges the remaining entity pairs into L for the next iteration until convergence.

4.2. Attribute-Based Model. In our scenario, the attributes of an entity consist of a number of key-value pairs, for example, < name:Michael >, where "name" is the attribute key, and "Michael" is the attribute value. For simplicity, a key-value pair is also called an attribute.

4.2.1. Attribute Extension. A critical problem of attribute representation is that some actual attributes may not be observed since they are not explicitly built or captured by the crawlers. Therefore, we first extend the attributes of both KGs by using the prealigned entity pairs. Typically, given a couple of aligned entities, if one entity has an attribute in a KG, the other KG's corresponding entity also has this attribute. Based on such an observation, we can add a key-value pair to one entity in a KG if its counterpart in the other KG has this key-value pair. Formally, the attributes of an entity e_i are denoted by $a_{e_i} = \{p_1, p_2, \ldots, p_j, \ldots\}$, where $p_j = <$ key$_j$: value$_j >$ is a key-value pair. For each entity e_i in KG 1, its attribute $a_{e_i}^{(1)}$ can be extended to $\tilde{a}_{e_i}^{(1)}$ by

$$\tilde{a}_{e_i}^{(1)} = a_{e_i}^{(1)} \cup \left\{ p_k \mid p_k \in e_j^{(2)}, p_k \notin e_i^{(1)}, \left(e_i^{(1)}, e_j^{(2)}\right) \in L \right\}. \tag{1}$$

Similarly, its counterpart $a_{e_j}^{(2)}$ in KG 2 can be extended into $\tilde{a}_{e_j}^{(2)}$.

4.2.2. Attribute Feature Representation. In multilingual KGs, the attributes are in different languages and cannot be directly compared. However, our observation shows the following:

(1) The occurrence frequencies of equivalent attribute pairs, that is, attribute keys, in multilingual KGs are approximately similar. For example, an entity representing a person in different KGs often has some equivalent attributes such as name, date of birth, and nationality. Although the texts that describe these attributes are multilingual, their frequency in different KGs is similar to the ratio of person entities to all KG entities.

(2) The values of an equivalent attribute pair in different KGs has the same data type. For example, both the English word "Michael" and the Chinese word "Mai Ke" are strings, and both "3.14" and "3.14159" are floating numbers.

Hence, this study represents the attributes of an entity by its attribute key frequencies and attribute value types. The description of entity attribute features can be illustrated briefly by a concrete example shown in Figure 2.

First, the attribute triples in each KG are merged into a set of key-value pairs, where the keys and values are then used to represent the frequency and the type features, respectively. The frequency F_{p_j} of an attribute p_j is a floating number ranging from 0 to 1, calculated as follows:

$$F_{p_j} = \frac{\text{Count}_{p_j}}{|E|}, \tag{2}$$

where Count_{p_j} is the occurrence number of attribute p_j in a KG and $|E|$ is the total number of entities in the KG. In this example, the frequencies of entity "Michael"'s nationality and birthdate are 0.2162 and 0.3351, respectively.

Second, we divide the frequency range (i.e., the interval of [0, 1]) into a sequence of small real intervals. Its frequency interval number can represent an attribute. In this paper, a proportional sequence is applied to split the frequency range. The interval for an attribute p_j can be computed by

$$\text{Interval}_{p_j} = \log_{(1+q)} \frac{F_{p_j}}{F_{\min i}}, \tag{3}$$

where q is the proportionality constant, and we fix $q = 0.001$ in this paper. $F_{\min i}$ is the least frequency of occurrence of an attribute in a KG. For example, "Nationality" and "Guo Ji" in Figure 2 are both in interval 2, although their frequencies are different. The parameter setting makes the interval more robust to small changes caused by noise, especially when more different frequency attributes are merged into one interval.

The value type of an attribute is its data type. Following previous work [27], this study distinguishes four kinds of data types, that is, Integer, Double, DateTime, and String. We encode the value type by a one-hot vector with the same dimension to the number of data types. For example, The
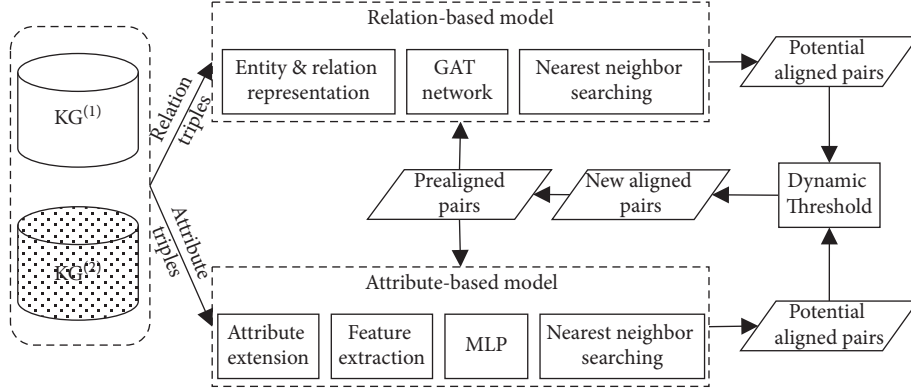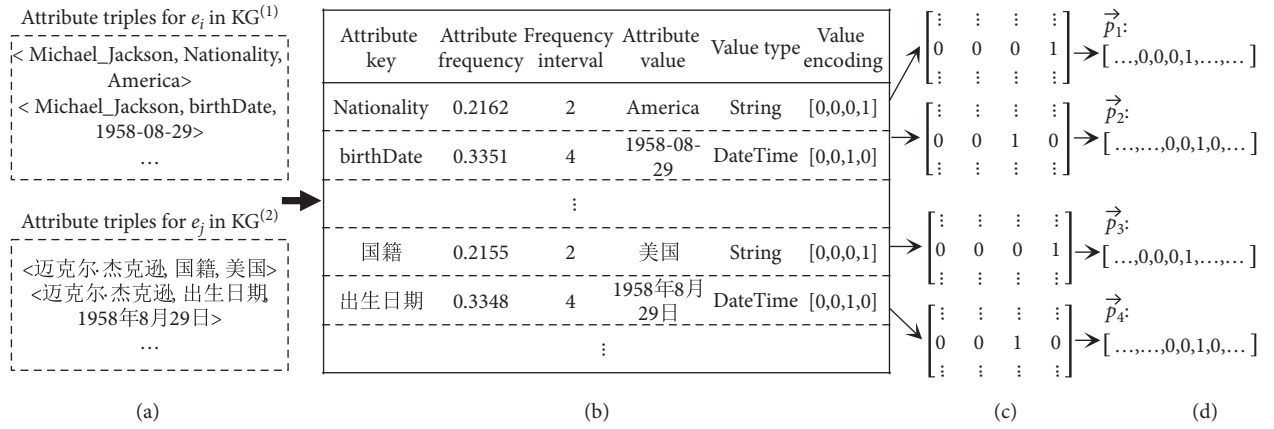
FIGURE 1: The overview of CAREA framework.

FIGURE 2: An example of entity attributes representation: (a) attribute triples in each KG are merged into a set of key-value pairs, (b) attribute keys and values are used to represent the frequency feature and type feature, respectively, (c) the frequency and type information are integrated into a sparse matrix, and (d) each attribute pair forms a feature vector.

codes for the attribute value "America" and "1958-08-29" are $[0, 0, 0, 1]$ and $[0, 0, 1, 0]$, respectively.

As explained in the above two steps, it is clear that the primary ideal of attribute feature representation is to integrate the representations of the frequency and the type of an attribute p_j. We combine the two representations into a sparse matrix as shown in Figure 2. Each row in the matrix denotes the value type, and the row index is the frequency interval number. On top of that, we reshape this matrix into a row feature vector $\overrightarrow{p_j}$. In this way, the attributes a_{e_i} for an entity e_i can be formed by the sum of its every attribute vectors $\overrightarrow{p_j}$ as

$$\overrightarrow{a_{e_i}} = \sum_{p_j \in a_{e_i}} \overrightarrow{p_j}. \tag{4}$$

To reduce noise, we use an indicator function $\mathbb{I}(\cdot)$ to transform the attribute vector of an entity into the following binary representation:

$$\mathbb{I}(x) = \begin{cases} 1, & \text{if } x \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The binary representation is averaged by the entity's neighbors as

$$\overrightarrow{a'_{e_i}} = \frac{1}{\left|\mathbb{N}_i^e\right|} \sum_{e_k \in \mathbb{N}_i^e} \mathbb{I}\left(\overrightarrow{a_{e_k}}\right), \tag{6}$$

where $\mathbb{N}_i^e$ denotes the neighboring entities of e_i. Then, a three-layer MLP transforms the attribute vectors of the two KGs into a uniform vector space, making the equivalent entities in different KGs close to each other. The MLP output is considered as the embedding of an entity, which is represented as $h_{e_i}^{\text{attr}}$. We use ReLU as an activation function in this paper. Batch normalization and dropout are added to increase performance. The details of the objective function are introduced in Section 4.4.

4.3. Relationship-Based Model. In KGs, there are various types of relations describing the role of entity associations that are crucial to aligning entities across KGs. Many previous works represent a relation by a transformation of the relationship connected entities. However, these methods bring the relationship too close to the entity [31]. Therefore, it will be difficult to capture the features of multiple relationships. Hence, this paper represents the entities and relations separately. Their combinations are adopted as the inputs to a graphical attention network (GAT) [34]. As a

result, the two KGs are embedded into a unified vector space so that the equivalent entities in different KGs are close to each other. This study treats relations as undirected; that is, (e_i, e_j) is equivalent to (e_j, e_i).

The idea of GAT is to calculate each entity's hidden representations in two KGs by focusing on their entity neighbors. GAT follows a self-attention strategy in its learning process. First, the embedding $h_{e_i} \in \mathbb{R}^{d \times 1}$ of each entity e_i and its connected relation $h_{r_j} \in \mathbb{R}^{d \times 1}$ are randomly initialized. This study sets the embedding dimension of entities and relations as the same d. Second, we average the entity e_i with its neighbors. Then, the entity embedding and the averaged embedding of entity connections are concatenated as the input to the GAT network as

$$h_{e_i}^{\text{in}} = \left[\frac{1}{|\mathbb{N}_i^e| + 1} \sum_{e_j \in \mathbb{N}_i^e \cup e_i} h_{e_j} \left\| \frac{1}{|\mathbb{N}_i^r|} \sum_{r_j \in \mathbb{N}_i^r} h_{r_j} \right. \right], \quad (7)$$

where $\mathbb{N}_i^e$ represents e_i's neighboring entities, $\mathbb{N}_i^r$ represents the set of relations that are outward from e_i, and notation $\|$ represents the concatenate operation. The attention coefficients can be calculated by

$$e_{ij} = a^T \left[h_{e_i}^{\text{in}} \left\| h_{e_j}^{\text{in}} \right. \right], \quad (8)$$

where e_{ij} indicates the weighted importance of neighboring e_j to e_i and $a \in \mathbb{R}^{2d \times 1}$ is the shared attention weight vector.

Different from the original GAT, there is no weight matrix W for the input feature in equation (8). In this study, all adjacent entities are normalized using softmax function and LeakyReLU nonlinearity with negative input slope $\alpha = 0.2$. Such normalization makes the coefficients between different nodes easy to compare, which can be denoted by

$$\alpha_{ij} = \text{Soft max}(e_{ij}) = \frac{\exp(\text{Leaky Re LU}(e_{ij}))}{\sum_{k \in \mathbb{N}_i^e} \exp(\text{Leaky Re LU}(e_{ik}))}. \quad (9)$$

Nonlinear ReLU is applied to the combination of participating neighbors. The operation yields the output features of each entity:

$$h_{e_i}^{\text{out}} = \text{Re LU}\left(\sum_{e_j \in \mathbb{N}_i^e} \alpha_{ij} h_{e_j}^{\text{in}} \right). \quad (10)$$

The stability of the training process is prepared by adopting a multihead mechanism. Specifically, K independent heads of attention execute the transformation of equation (10). Then, the averaged features result in the following output:

$$h_{e_i}^{\text{out}} = \text{Re LU}\left(\frac{1}{K} \sum_{k=1}^{k=K} \left[\sum_{e_j \in \mathbb{N}_i^e} \alpha_{ij}^k h_{e_j}^{\text{in}} \right] \right), \quad (11)$$

where k is the indicator of heads and α_{ij}^k represents the attention coefficient in the k-th head. This study also expands the attention mechanism to multihop neighboring level information by adding more layers, thus creating a

more global-aware representation of the KG. Let $h_{e_i}^{\text{out}(0)}, \ldots, h_{e_i}^{\text{out}(l)}$ be the output features of e_i from 0-th (input features) to l-th layer. We concatenate them together to obtain the final output features $h_{e_i}^{\text{struc}}$ of entity e_i as

$$h_{e_i}^{\text{struc}} = \left[h_{e_i}^{\text{out}(0)} \|, \ldots, \| h_{e_i}^{\text{out}(l)} \right]. \quad (12)$$

4.4. Objective Function.
As was mentioned in the previous section, the two models both provide embeddings of the entities for two KGs from different views. This section uses the same objective function to optimize both of them. Following the previous work [33], Manhattan Distance is employed to be the similarity measure. The similarity of $e_i \in G^{(1)}$ and $e_j \in G^{(2)}$ in the joint vector space can be calculated by

$$s\left(e_i^{(1)}, e_j^{(2)}\right) = \left| h_{e_i}^{(\text{attr/struc})} - h_{e_j}^{(\text{attr/struc})} \right|. \quad (13)$$

All similar entities in $G^{(2)}$ should be calculated using the same method to find the entity e_i's counterpart. The nearest one is chosen as e_i's equivalent. On top of that, we adopt the following margin-based loss function since it ensures the balance between positive and negative samples and ensures the lower scores for positive ones; that is,

$$L = \sum_{(e_i, e_j) \in L} \left\{ \left[s(e_i, e_j) - s(e_i - e_j') + \gamma \right]_+ \right. \\ \left. + \left[s(e_i, e_j) - s(e_i' - e_j) + \gamma \right]_+ \right\}, \quad (14)$$

where $[\cdot]_+$ represents $\max(\cdot, 0)$ and γ is a hyperparameter of margin. e_i' and e_j' are the negative counterpart of e_i and e_j, respectively. In this work, the entities in $G^{(1)}$ and $G^{(2)}$ are randomly selected as negative counterparts. Adam [35] is adopted to minimize the loss function.

4.5. Cotraining Algorithm.
In this study, the cotraining process of the attribute-based model f_{attr} and relation-based model f_{struc} is conducted iteratively. Two components alternately take turns to train and predict new potential aligned entity pairs at each iteration until either of the two parts no longer obtain new pairs. Such a prediction is based on the cosine similarity of entities in the united vector space. A new pair sourced from a KG is suggested by searching the nearest neighbor (NN) in the other KG. It is worth noting that, in most cases, the NNs are asymmetric. For example, although e_i in $G^{(1)}$ is the most similar entity to e_j in $G^{(2)}$, there may be another entity in $G^{(2)}$ closer to e_i. Thus, the newly predicted entity pairs should be bidirectional nearest neighbors.

4.5.1. Dynamic Similarity Threshold.
We further evaluate the predicted potential aligned entity pairs by dynamically adjusting the threshold in each iteration. That is to say, only the entity pairs, whose cosine similarity falls within a certain threshold τ, are populated into the aligned pair set L. As higher similarity threshold implies higher precision, we set higher thresholds for earlier iterations. However, it may also

limit the capability of the model to propose a sufficient number of aligned entity pairs. Thus, lower thresholds are taken for later iterations. The design of the threshold function is alternative. In this paper, we design a linear threshold function as

$$\tau(n) = \alpha - \delta(n), \qquad (15)$$

where $\alpha \in (0, 1)$ is the initial threshold, $n \in \{0, 1, 2, \ldots\}$ is the iteration number, and $\delta \in (0, 1)$ is the coefficient controlling the changing rate of each iteration. In order to control the precision of each component model, we set different threshold parameters for different models in our experiments. The detailed cotraining procedure of CAREA is given in Algorithm 1.

5. Experiments

5.1. Datasets. This section applies a popular public dataset DBP-15K [27] for entity alignment to evaluate the performance of the approach CAREA. DBP-15 K contains three cross-lingual subsets built from DBpedia: DBP$_{ZH-EN}$, DBP$_{JA-EN}$, and DBP$_{FR-EN}$. Each of the three subsets contains two KGs in different languages, for example, DBP$_{FR-EN}$ for French and English. Their statistics are displayed in Table 1.

5.2. Experiment Settings. Following the previous work [29], we adopt two evaluation metrics: (1) Hits @ k: the proportion of correctly aligned entities ranked in the top k. (2) Mean Reciprocal Rank (MRR): the average of the reciprocal ranks of results. Higher Hits @ k and MRR scores indicate better alignment performance. The two metrics can be calculated as follows:

$$\text{score}(x) = \begin{cases} 1, & \text{if pos}(x) \le k, \\ 0, & \text{otherwise}, \end{cases}$$

$$\text{Hits}(x) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \text{score}(x), \qquad (16)$$

$$\text{MRR}(x) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \frac{1}{\text{pos}(x)},$$

where $\text{pos}(\cdot)$ is the position of tested entity pair in the returned list, $\mathcal{T}$ is the set of tested entity pairs, and Hits @ 1 and Hits @ 10 are adopted in our experiments.

This research is compared with other baselines by the same evaluation metrics and the dataset's splitting method. The experiment randomly splits 30% of the prealigned entity pairs as training data, while the rest 70% for testing. The average score of both alignment directions (e.g., ZH ⟶ EN vs. EN ⟶ ZH) is reported by considering the asymmetric of the nearest relation across KGs. Each experiment instance is run five times independently. Their average performances are considered as the final results. The same settings for experiment models are applied unless otherwise stated. The hidden dimensions for attributes, entities, and relations are the same: $d = 100$. The margin parameter γ, dropout rate,

and learning rate of Adam are 3, 0.3, and 0.005, respectively. For the relation-based model f_{struc}, we fix the number of attention heads $K = 2$ and GAT layer's depth $l = 2$. For the cotraining process, we take the results of CAREA's third iteration as its final performance. On the other hand, the threshold coefficients for both model components are the same as $\delta = 0.05$. The initial thresholds α for f_{attr} and a_{struc} are empirically set to 0.95 and 0.9, respectively.

5.3. Baselines. To demonstrate the advantage of our method, we compared it with the following baselines:

(i) MtransE [24]: MtransE is a structure-based model for multilingual KG embeddings, to provide a simple and automated solution. The model characterizes monolingual relations and deploys three different techniques to represent cross-lingual transitions, namely, axis calibration, translation vectors, and linear transformations.

(ii) JAPE [27]: JAPE is an attribute-preserving embedding model that incorporates the relation and attribute embeddings for entity alignment.

(iii) GCN-Align [33]: GCN-Align employs GCNs to learn embeddings from both the structure and attribute information of entities for cross-lingual KG alignment.

(iv) BootEA [26]: BootEA adopts a bootstrapping strategy, which iteratively labels potential entity alignments as training data and leverages it for learning alignment-oriented embeddings.

(v) MuGNN [28]: MuGNN learns alignment-oriented KG embeddings by robustly encoding two KGs via KG completion and entities pruning.

(vi) NAEA [29]: NAEA incorporates neighborhood subgraph-level information of entities and designs a neighborhood-aware attentional representation mechanism on multilingual KGs.

The performances of the above baselines come from the reported results in their papers. We also evaluate the effectiveness of the component models of our approach, including the following:

(i) Attribute-based model: the model, denoted as CAREA-a, ignores structure embedding components to assess the effect of the attribute embedding. In other words, the attribute features are only used to align entities without a cotraining strategy.

(ii) Structure-based model: we also estimate the component's performance of network structure embedding, which ignores the attribute features except for the structure ones to align entities. The model is similarly denoted as CAREA-s.

5.4. Experiment Results

5.4.1. Overall Performance. All the comparison methods can be treated under two groups according to feature categories. One is purely based on KG structures, including MTransE,

Input: Two KGs to be aligned $G^{(1)}$ and $G^{(2)}$, a set L of prealigned entity pairs, and the parameters $\alpha_1, \alpha_2, \delta_1, \delta_2$ of threshold function.
Output: The parameters of f_{attr} and f_{struc}.
(1) Initial iteration number $n = 0$;
(2) **repeat**
(3)   Reinitialize f_{attr} and f_{struc};
(4)   Train f_{struc} based on L;
(5) $S \leftarrow NN(f_{struc}, G^{(1)}, G^{(2)}) \cap NN(f_{struc}, G^{(2)}, G^{(1)})$;
(6) $\tau_1 = \alpha_1 - \delta_1 \times n$
(7) $S' = \{(e_i, e_j) | similarity t (e_i, e_j) < \tau_1, (e_i, e_j) \in \quad S\}$;
(8) $L \leftarrow L \cup S$
(9)   Train f_{attr} based on L;
(10) $S \leftarrow NN(f_{attr}, G^{(1)}, G^{(2)}) \cap NN(f_{attr}, G^{(2)}, G^{(1)})$;
(11) $\tau_2 = \alpha_2 - \delta_2 \times n$;
(12) $S' = \{(e_i, e_j) | similarity t (e_i, e_j) < \tau_2, (e_i, e_j) \in S\}$;
(13) $L \leftarrow L \cup \quad S$
(14) $n \leftarrow n + 1$
(15) **until** $S' = \phi$;

ALGORITHM 1: Cotraining of CAREA.

BootEA, MuGNN, NAEA, and CAREA-s. The other leverages both entity attributes and relations for entity alignment, including JAPE, GCN-Align, and CAREA. Table 2 summarizes the overall results of all compared methods on the three datasets.

In the structure-based group, our model CAREA-s performs better than MTransE by at least 27.3% in terms of Hits@1 on three datasets, which also resulted in a greater score than MuGNN by at least 8.4%. The comparison results have demonstrated the effectiveness of our structure-based approach. Further tests in other group revealed that CAREA outperforms JAPE and GCN-Align by at least 28.6% across all datasets on Hits@1. The result demonstrates our approach's superiority by leveraging both entity attributes and KG structures for entity alignment. Finally, CAREA ranked the best over all the competing approaches across all datasets. For example, the performance of CAREA is better than NAEA and BootEA by at least 3.9% and 6.0% in terms of Hits@1, respectively.

Another proposed component model CAREA-a is excluded from the above comparisons since it is the only one that relies solely on attribute information to align entities. It achieves lower scores. For example, when CAREA-a were stimulated with Hits@1 and Hits@10 on DBP_{ZH-EN}, lower scores of 22.1% and 51.8% were reported. This is mainly because of the heterogeneity across multilingual KGs or the KGs may not be explicitly constructed or captured by the crawlers. Although CAREA-a does not perform as well as that of structure-based approaches, it provides another view for entity alignment and improves the performance of our method in the KG alignment task.

5.4.2. Effects of Cotraining Algorithm. This part confirms the achievements of the approach CAREA by presenting each iteration of the cotraining process. The result is shown in

Figure 3. Its trend reveals that there has been a gradually similar increase in terms of all evaluation metrics, which is verified on all the three datasets of our two-component models. The iterative cotraining algorithm significantly improves the performance, with at least an absolute increase of 10.5% Hits@1 across all experiment datasets. Both the attribute f_{attr} model and the structure model f_{struc} get enhanced from each iteration. After 3 to 4 iterations, the component model performance becomes stable.

5.4.3. Parameter Sensitivity Analysis. This part investigates the parameter sensitivity of the proposed CAREA on three primary parameters: (1) the proportion of the prealigned entity pairs, (2) the feature dimension d, and (3) the margin parameter γ of the proposed objective function.

Sensitivity to data proportions. We run CAREA by the training proportions from 10% to 50% with a step size of 10%. Figure 4 illustrates the change of Hits@k concerning different proportions. The shown results on all the datasets become better with the proportion increase following our expectations. The amount of data is a significant factor that more training data can provide more extended information to overlay the cross-lingual KGs. Figure 4 shows that CAREA performed encouragingly when using only 10% of the aligned entities as training data. For example, Hits@1 and Hits@10 on DBP_{ZH-EN} are 56.2% and 82.4%, respectively. Therefore, CAREA is expected to be well adapted to annotate constrained scenarios.

Sensitivity to the feature dimension d: Figure 5 depicts the sensitivity of the model performance on different feature dimensions. The model performance of CAREA

TABLE 1: The statistics of the datasets used in the experiment.

Datasets		#Entities	#Relations	Attributes	Attr. triples	Rel. triples
DBP$_{ZH-EN}$	Chinese	19,**388**	2,830	8,113	153,929	379,684
	English	19,572	2,317	7,173	237,674	567,755
DBP$_{JA-EN}$	Japanese	19,814	2,043	5,882	164,373	354,619
	English	19,780	2,096	6,066	233,319	497,230
DBP$_{FR-EN}$	French	19,661	1,379	4,547	192,191	528,665
	English	19,993	2,209	6,422	278,590	576,543
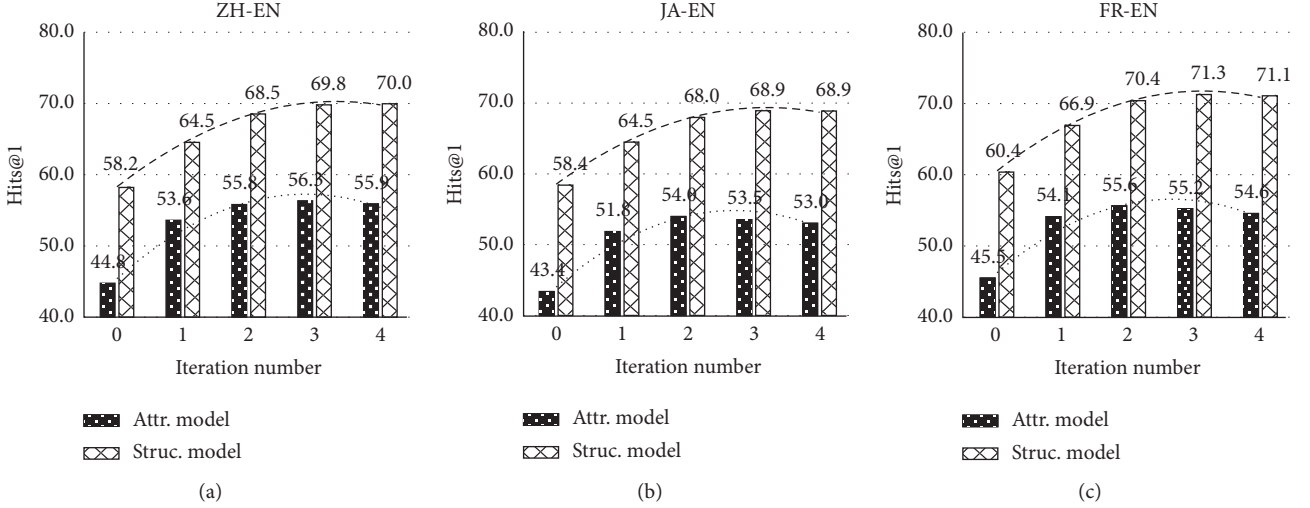
FIGURE 3: Hits@1 performances of the two-component models in each iteration. (a) DBP$_{ZH-EN}$. (b) DBP$_{JA-EN}$. (c) DBP$_{FR-EN}$.
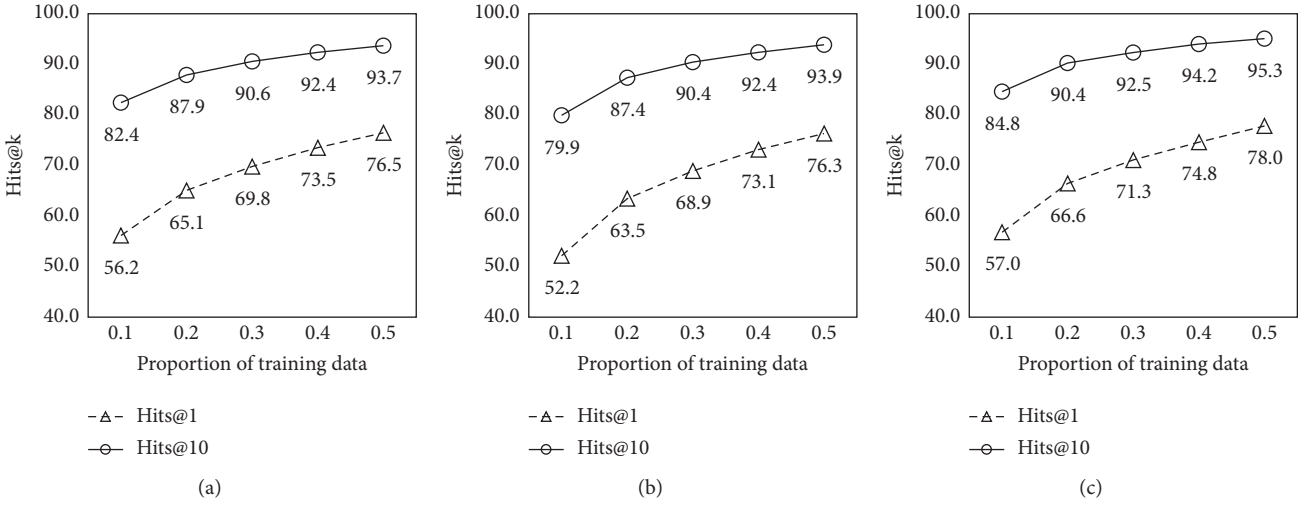
FIGURE 4: Hits @ k on entity alignment with respect to the proportion of training data. (a) DBP$_{ZH-EN}$. (b) DBP$_{JA-EN}$. (c) DBP$_{FR-EN}$.

keeps stable with the data dimension increases in all datasets. It is proved that a high-dimensional feature space is helpful to preserve entity information and improve the entity alignment performance. A larger dimension necessarily consumes more computing resources. Hence, we chose $d = 100$ to weigh efficiency against effectiveness.

Sensitivity to the margin γ: the model performance produced by setting different margin parameter γ (from 1 to 4) of the objective function is shown in Figure 6. The performances become steady when $\gamma \geq 2$ with at most 2.5% range from all datasets. Therefore, CAREA can keep stable when γ varies within a reasonable range.
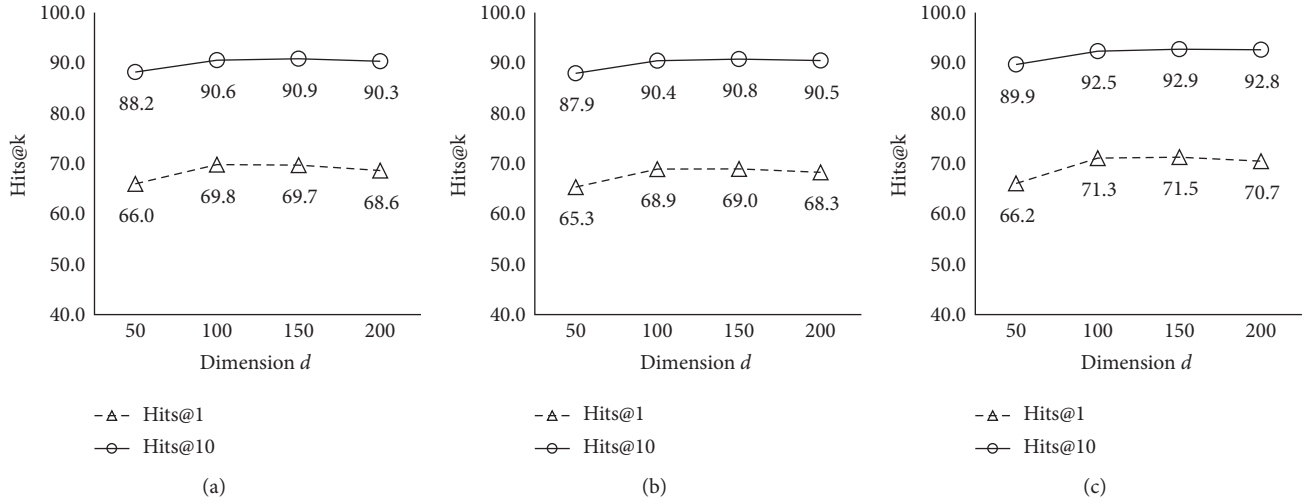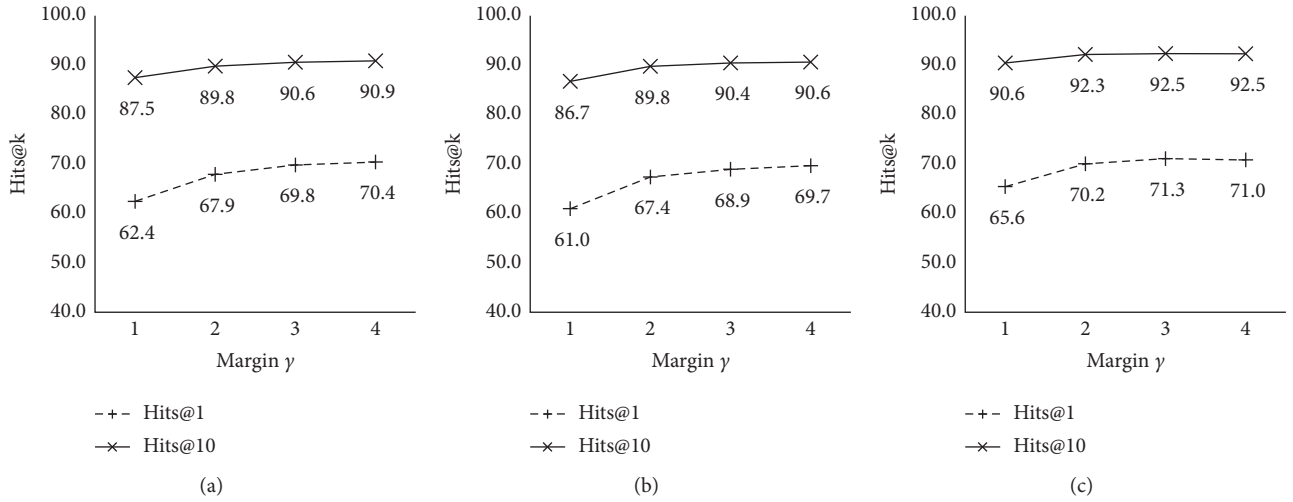
FIGURE 5: CAREA performance with respect to the feature dimension d. (a) DBP$_{\text{ZH-EN}}$. (b) DBP$_{\text{JA-EN}}$. (c) DBP$_{\text{FR-EN}}$.

FIGURE 6: CAREA performance with respect to the margin parameter γ. (a) DBP$_{\text{ZH-EN}}$. (b) DBP$_{\text{JA-EN}}$. (c) DBP$_{\text{FR-EN}}$.

TABLE 2: Comparison with the baseline methods.

Method	DBP$_{\text{ZH–EN}}$			DBP$_{\text{JA–EN}}$			DBP$_{\text{FR–EN}}$		
	Hits @ 1	Hits @ 10	MRR	Hits @ 1	Hits @ 10	MRR	Hits @ 1	Hits @ 10	MRR
MTransE	30.8%	61.4%	0.364	27.9%	57.5%	0.349	24.4%	55.6%	0.335
JAPE	41.2%	74.5%	0.490	36.3%	68.5%	0.476	32.4%	66.7%	0.430
GCN-Align	41.3%	74.4%	0.549	39.9%	74.5%	0.546	37.3%	74.5%	0.532
MuGNN	49.4%	84.4%	0.611	50.1%	85.7%	0.621	49.5%	87.0%	0.621
BootEA	62.9%	84.8%	0.703	62.2%	85.4%	0.701	65.3%	87.4%	0.731
NAEA	65.0%	86.7%	0.720	64.1%	87.3%	0.718	67.3%	89.4%	0.752
CAREA-a	22.1%	51.8%	0.320	18.4%	46.4%	0.277	16.9%	45.4%	0.264
CAREA-s	58.2%	87.5%	0.685	58.5%	87.8%	0.689	60.5%	90.5%	0.711
CAREA	69.8%	90.6%	0.772	68.9%	90.4%	0.766	71.3%	92.5%	0.789

6. Conclusion and Future Work

The purpose of the present research was to investigate the cross-lingual entity alignment problem in KGs. This study has constructed a cotraining based approach CAREA to learn entity embeddings from two independent views of knowledge (relationships and attributes). CAREA is innovatively constructed as a two-component model f_{attr} and f_{struc}, which can extract the attribute and the relation information, respectively. In each iteration, both models alternately take turns of the train-and-predict process, which gradually improves each model's performance. Experiments on three popular datasets confirm the effectiveness and superiority of CAREA on the entity alignment task. The insights of model construction gained from this study may be of assistance to complex multilanguage and cross-domain knowledge organization and analysis. Future work seeks to extend the CAREA method to other applications, such as link prediction, information extraction, and entity classification.

Data Availability

The data used to support the findings of this study are publicly available at https://github.com/ChenBaiyang/CAREA or https://github.com/nju-websoft/JAPE.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] C. Bizer, J. Lehmann, G. Kobilarov et al., "DBpedia—a crystallization point for the web of data," *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: a large ontology from wikipedia and WordNet," *Journal of Web Semantics*, vol. 6, no. 3, pp. 203–217, 2008.

[3] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: an open multilingual graph of general knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017*, pp. 4444–4451, San Francisco, CA, USA, February 2017.

[4] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 92–100, Association for Computing Machinery, Madison, WI, USA, 1998.

[5] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp. 86–93, Association for Computing Machinery, McLean, VA, USA, 2000.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NIPS 2013*, pp. 2787–2795, 2013.

[7] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the 28th AAAI conference on artificial intelligence, AAAI 2014, 26th innovative applications of artificial intelligence conference, iaai 2014 and the 5th symposium on educational advances in artificial intelligence, EAAI 2014*, pp. 1112–1119, AI Access Foundation, Québec City, Canada, July 2014.

[8] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015*, pp. 2181–2187, AI Access Foundation, Austin, Texas USA, January 2015.

[9] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP 2015*, pp. 687–696, Association for Computational Linguistics, Beijing, China, July 2015.

[10] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proceedings of the 15th International Conference on Extended Semantic Web Conference, ESWC 2018*, pp. 593–607, Springer Verlag, Heraklion, Greece, June 2018.

[11] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pp. 809–816, Association for Computing Machinery, Bellevue, WA, USA, June 2011.

[12] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NIPS 2013, Air Force Office of Scientific Research (AFOSR)*, Tahoe, CA, USA, December 2013.

[13] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable learning of logical rules for knowledge base reasoning," in *Proceedings of the 31st Annual Conference on Neural Information Processing Systems, NIPS 2017*, pp. 2320–2329, Neural information processing systems foundation, Long Beach, CA, USA, December 2017.

[14] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence,*

AAAI 2018, pp. 1811–1818, AAAI press, New Orleans, LA, USA, February 2018.

[15] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, pp. 4710–4723, Association for Computational Linguistics (ACL), Florence, Italy, 2019.

[16] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp. 705–714, Association for Computational Linguistics (ACL), Lisbon, Portugal, September 2015.

[17] Y. Lin, Z. Liu, and M. Sun, "Knowledge representation learning with entities, attributes and relations," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 2866–2872, International Joint Conferences on Artificial Intelligence, Palo Alto, CA, USA, July 2016.

[18] D. Krompac, S. Baier, and V. Tresp, "Type-constrained representation learning in knowledge graphs," in *Proceedings of the 14th International Semantic Web Conference, ISWC 2015*, pp. 640–655, Springer Verlag, Bethlehem, PA, USA, October 2015.

[19] P. Ristoski and H. Paulheim, "RDF2Vec: RDF graph embeddings for data mining," in *Proceedings of the 15th International Semantic Web Conference, ISWC 2016*, pp. 498–514, Springer Verlag, Kobe, Japan, October 2016.

[20] M. Cochez, P. Ristoski, S. P. Ponzetto, and H. Paulheim, "Global RDF vector space embeddings," in *Proceedings of the 16th International Semantic Web Conference, ISWC 2017*, pp. 190–207, Springer Verlag, Vienna, Austria, October 2017.

[21] W. Hu, J. Chen, and Y. Qu, "A self-training approach for resolving object coreference on the semantic web," in *20th International Conference on World Wide Web*, pp. 87–96, Association for Computing Machinery, New York, NY, USA, 2011.

[22] F. M. Suchanek, S. Abiteboul, and P. Senellart, "Paris," *Proceedings of the VLDB Endowment*, vol. 5, no. 3, pp. 157–168, 2011.

[23] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani, "SiGMa: simple greedy matching for aligning large knowledge bases," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 572–580, Association for Computing Machinery, New York, NY, USA, 2013.

[24] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1511–1517, Melbourne, Australia, August 2017.

[25] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4258–4264, AAAI Press, Melbourne, Australia, August 2017.

[26] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4396–4402, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, July 2018.

[27] Z. Sun, W. Hu, and C. Li, "Cross-lingual entity alignment via joint attribute-preserving embedding," in *Lecture Notes in Computer Science, The Semantic Web–ISWC 2017*, Springer International Publishing, New York, NY, USA, 2017.

[28] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li, and T.-S. Chua, "Multi-channel graph neural network for entity alignment," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1452–1461, Florence, Italy, August 2020.

[29] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "Neighborhood-aware attentional representation for multilingual knowledge graphs," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1943–1949, AAAI Press, Macao, China, August 2019.

[30] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao, "Relation-aware entity alignment for heterogeneous knowledge graphs," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5278–5284, Macao, China, August 2019.

[31] X. Mao, W. Wang, H. Xu, M. Lan, and Y. Wu, "MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 420–428, Association for Computing Machinery, Houston, TX, USA, February 2020.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017.

[33] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, "Cross-lingual knowledge graph alignment via graph convolutional networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 349–357, Association for Computational Linguistics, Brussels, Belgium, October 2018.

[34] P. Velikovi, A. Casanova, P. Lio, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, ICLR, Vancouver, Canada, Febrary 2018.

[35] D. P. Kingma and J. L. Ba, "ADAM: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings*, San Diego, CA, USA, May 2015.