

A Localized Geometric Method to Match Knowledge in Low-dimensional Hyperbolic Space

Bo Hui

Auburn University
bohui@auburn.edu

Tian Xia

Auburn University
tianxia@auburn.edu

Wei-Shinn Ku

Auburn University
weishinn@auburn.edu

Abstract

Matching equivalent entities across Knowledge graphs is a pivotal step for knowledge fusion. Previous approaches usually study the problem in Euclidean space. However, recent works have shown that hyperbolic space has a higher capacity than Euclidean space and hyperbolic embedding can represent the hierarchical structure in a knowledge graph. In this paper, we propose a localized geometric method to find equivalent entities in hyperbolic space. Specifically, we use a hyperbolic neural network to encode the lingual information of entities and the structure of both knowledge graphs into a low-dimensional hyperbolic space. To address the asymmetry of structure on different KGs and the localized nature of relations, we learn an instance-specific geometric mapping function based on rotation to match entity pairs. A contrastive loss function is used to train the model. The experiment verifies the power of low-dimensional hyperbolic space for entity matching and shows that our method outperforms the state of the art by a large margin.

1 Introduction

Knowledge graph (KG) is knowledge base that uses a graph-structured topology to integrate entities, relations, and metadata. Real-world KGs such as DBpedia, Wikidata, and Yago benefit a variety of downstream applications such as question answering (Cui et al., 2017), and fact checking (Huynh and Papotti, 2019). In general, a KG is constructed from one single knowledge base or built in one single language. Thus it is impractical to reach full coverage of the domain (Zhao et al., 2020). To increase the completeness of the knowledge base, a conventional approach is fusion of multiple KGs. One pivotal step for fusion is to align equivalent entities across different KGs.

Conventional entity alignment approaches mainly compare symbolic features of entities (Lacoste-Julien et al., 2013) or reason the co-

relations by ontology matching (Jiménez-Ruiz and Grau, 2011). With the prosperity of node embedding (Grover and Leskovec, 2016), recent works favor learning entity embeddings for alignment and compare entities using embedding distance metrics. Existing embedding methods for entity alignment can be classified into three types: attribute-based (Sun et al., 2017), relation-based (Chen et al., 2017; Mao et al., 2020) and graph-based (Wang et al., 2018; Sun et al., 2020b).

However, these embedding-based works study the problem in Euclidean space, where the embeddings are Euclidean vectors. Recent research has proven that Euclidean space does not provide the most powerful geometrical representations for complex data that exhibit a highly non-Euclidean latent anatomy (Bronstein et al., 2017; Hui and Ku, 2022). To tackle this challenge, a variety of remarkable embedding methods have been developed to represent the data in hyperbolic space. The distinctive feature of hyperbolic spaces enables us to embed hierarchical data while preserving the latent hierarchical structure (Nickel and Kiela, 2017).

In this paper, we propose to solve the entity alignment problem in *hyperbolic space*. Since entity alignment is a downstream task for embedding, how to use the hyperbolic embedding to match the entities is a challenge. Furthermore, all operations in neural networks such as vector addition, matrix-vector multiplication, and vector inner product are defined in Euclidean space. Therefore, existing neural network models are not applicable anymore for hyperbolic embeddings. To address these challenges, we use a hyperbolic version of neural networks. Specifically, we utilize a hyperbolic graph neural network model to learn the low-dimensional hyperbolic embeddings for entities on two KGs respectively. Two mapping functions are used to implement the initialization, attention-based aggregation, and reduction of dimension in the model. The pre-trained semantic embeddings are projected

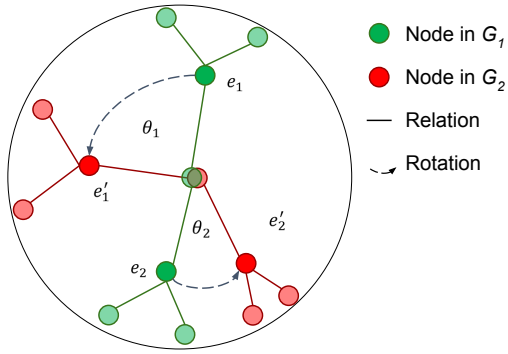


Figure 1: KGs in 2-dimensional Poincaré disk.

into the hyperbolic space as the entity features. We consider the output of the hyperbolic neural network model as the final low-dimensional hyperbolic embeddings for entities. Then the next task is to map embeddings between two KGs.

Existing works use a *unified global* map function to match entities. However, The asymmetry of structure on different KGs makes it difficult to learn a unified relationship for all pairs. The reason behind the varied relations is the heterogeneous nature of data sources for KGs. For example, there are 30,291 inner-relations between 15,000 entities in DBpedia KG. However, there are only 26,638 inner-relations between these entities in Yago KG. As a result, the structure of one DBpedia sub-graph may be the same as its counterpart in Yago KG, where the structure of another sub-graph may be different from its counterpart. To address the localized relations between entities, we propose an *instance-based* geometric mapping function where *local* parameters are learned from its embeddings for each entity. In hyperbolic space, more generic/ambiguous nodes (e.g., root in a KG) tend to be placed closer to the origin while moving more specific objects (nodes at low levels) towards the boundary. Figure 1 shows the embeddings of two KGs in a 2-dimensional Poincaré disk. We can see that the nodes at low levels will be closer to the original point while nodes at higher levels are placed towards the boundary. This motivates us to design a novel geometric mapping function based on *rotation* to match two entities across KGs. Ideally, after the rotation, the entity will overlap with the equivalent entity in the hyperbolic space. Instead of learning a unified rotation function for all entities, we use instance-based rotation functions. As shown in Figure 1, after the rotation of embedding for e_1 through θ_1 , e_1 will overlap with its corresponding entity e'_1 . However, the angle between embeddings of e_2 and e'_2 is θ_2 , which is

totally different from θ_1 .

To train the model, we minimize the hyperbolic distance after mapping between a pair of aligned entities and push negative samples away from the target one. For each entity, we find the aligned entity by searching for the nearest neighbor in terms of hyperbolic distance. Our novelty over existing works can be summarized as:

- We solve the entity alignment problem in hyperbolic space instead of Euclidean space to capture hierarchical structures of KGs.
- We propose a hyperbolic geometric mapping function to address the non-linear distance ratio with respect to radius in hyperbolic space.
- Instead of using a unified global mapping function for all entities, we learn the localized parameters for each entity to address the asymmetry of structure on different KGs.

2 Related Work

The majority of existing entity matching methods rely on KG embeddings (Sun et al., 2020c). According to the KG embeddings approaches, these models can be roughly categorized into three groups: relation-based, attributes-based, and graph-based models. Relation-based models mainly employ the translational methods (Bordes et al., 2013) to learn the embedding based on relationship triples. IPTransE (Zhu et al., 2017) is an entity alignment model based on translation. It encodes both entities and relations into a unified low-dimensional semantic space. MTransE (Chen et al., 2017) encodes entities and relations of each KG in a separated embedding space. BootEA (Sun et al., 2018) leverages the bootstrapping idea to iteratively label likely alignment. RSNs (Guo et al., 2019) feeds the relational paths into recurrent neural networks to learn embeddings. To increase the robustness, MultiKE (Zhang et al., 2019) unifies multiple views of entities and embeds entities with several combination strategies. Attributes-based models consider the correlations among attributes of entities. For example, JAPE (Sun et al., 2017) assumes that similar entities should have similar correlated attributes. AttrE (Trisedya et al., 2019) exploits large numbers of attribute triples and models the various types of attribute triples to generate attribute embeddings. Then the embedding shifts two KGs into the same space by computing the similarity.

With the prosperity of graph neural networks (Kipf and Welling, 2017; Hui et al., 2020; Jiang et al., 2022), many works propose to utilize graph convolutional networks to model the structure of KG. GCNAlign (Wang et al., 2018) trains GCNs to embed entities of each KG into a unified vector space. RDGCN (Wu et al., 2019) further incorporates relation information in KG and captures neighboring structures via dual-Graph convolutional network. AliNet (Sun et al., 2020b) aims to mitigate the non-isomorphism of neighborhood structures in an end-to-end manner and controls the aggregation of both direct and distant neighborhood information using a gating mechanism. RREA (Mao et al., 2020) abstracts existing entity alignment methods into a unified framework and derives two key criteria for an ideal transformation operation. RNM (Zhu et al., 2021) is a relation-aware neighborhood matching model. It utilizes neighborhood matching to enhance the entity alignment and uses an iterative framework to leverage the positive samples and the relation alignment in a semi-supervised manner. Dual-AMN (Mao et al., 2021) uses an encoder to model both intra-graph and cross-graph information. EASY (Ge et al., 2021) removes the labor-intensive pre-processing by fully discovering the name information provided by the entities themselves and jointly fuses the features captured by the names of entities and the structural information of the graph. ActiveEA (Liu et al., 2021) introduces Active Learning to reduce the cost of labeling and annotation. Temporal KG is also studied to match time-aware entities (Xu et al., 2021). HMEA (Guo et al., 2021) utilizes visual information to learn image embeddings. It combines the structure and visual representations in the hyperbolic space to predict alignment results.

HyperKA (Sun et al., 2020a) also aligns entities across KGs in hyperbolic space. However, HyperKA directly aggregates neighborhood information in hyperbolic space and fails to leverage the power of hyperbolic neural networks (e.g., dimensionality reduction and attention mechanism). Furthermore, it uses a unified linear transformation function where we use a localized geometric method; thus HyperKA ignores the non-linear distance ratio with respect to the radius, the isometry of hyperbolic geometry and the locality of mapping. Lastly, it randomly associates each entity with a vector. Instead, we associate the entity with a pre-trained semantic embedding. Besides entity align-

ment, tensor completion (Harshman et al., 1970; Hui et al., 2022) is another method to increase the completeness of the knowledge base.

3 Preliminaries

3.1 Problem Formulation

We use $G = (E, R, T)$ to represent a KG, where E and R are the sets of entities and relations in the KG. Let T be the set of triples, each of which is (e_h, r, e_t) , including the head entity $e_h \in E$, the tail entity $e_t \in E$ and the relation r between e_h and e_t . In the entity alignment problem, we are given two KGs: $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$. The set of known aligned entity pairs across G_1 and G_2 is defined as: $S = \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2\}$, where e_1 and e_2 are equivalent to each other. Our goal is to find more 1-to-1 alignments across two KGs G_1 and G_2 .

3.2 Hyperbolic Geometry

Here we briefly present some basic knowledge in Hyperbolic geometry. Hyperbolic space is a complete simply connected Riemannian manifold with constant negative curvature. There are five isometric models for hyperbolic space: half-space, Poincaré, hemisphere, Klein, and 'Loid. In this paper, we use the d -dimensional Poincaré ball model which is most popular in machine learning:

$$B^{d,c} = \{\mathbf{x} \in R^d : \|\mathbf{x}\|^2 < \frac{1}{c}\}, \quad (1)$$

where $-c(c > 0)$ is the negative curvature. Different from Euclidean addition of two vectors, the Möbius addition of \mathbf{x} and \mathbf{y} in hyperbolic space $B^{d,c}$ is defined as:

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \quad (2)$$

Then the hyperbolic distance between \mathbf{x} and \mathbf{y} in $B^{d,c}$ in the manifold is given by:

$$d_c(\mathbf{x}, \mathbf{y}) = (1/\sqrt{c})\text{arcosh}(-c\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{M}}), \quad (3)$$

where $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ denotes the Minkowski inner product.

3.3 Hyperbolic KG Embedding

A distinctive property of hyperbolic space is that the circle circumference and disc area grow exponentially with respect to radius, which allows hyperbolic embeddings to represent hierarchical structures. Specifically, given three points: the origin \mathbf{o} , \mathbf{x} and \mathbf{y} with $\|\mathbf{x}\| = \|\mathbf{y}\| = r (x \neq y)$, we depict the hyperbolic distance ratio $\frac{d_c(\mathbf{x}, \mathbf{y})}{d_c(\mathbf{x}, \mathbf{o}) + d_c(\mathbf{o}, \mathbf{y})}$ in

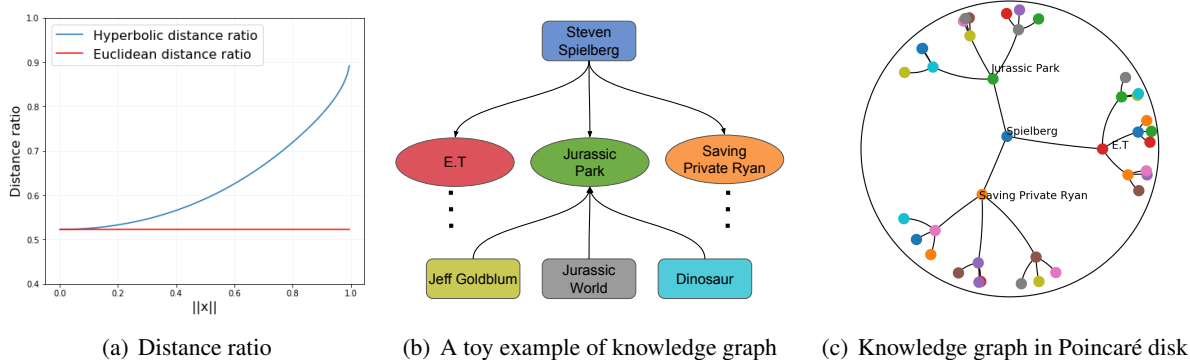


Figure 2: Hyperbolic space

Figure 2(a). Compare with Euclidean space where the distance ratio $\frac{d_e(\mathbf{x}, \mathbf{y})}{d_e(\mathbf{x}, \mathbf{o}) + d_e(\mathbf{o}, \mathbf{y})}$ ($d_e(\cdot)$ represents Euclidean distance) is constant, the hyperbolic distance ratio approaches to 1 exponentially as $r \rightarrow 1$. Equivalently, the shortest path from \mathbf{x} to \mathbf{y} is almost the same as the path through the origin as $r \rightarrow 1$. It is analogous to the property of tree data structure in which the shortest path between two sibling nodes is the path through their parent (Sala et al., 2018).

KGs often exhibit hierarchical structures and the number of nodes grows exponentially as the level increases. Figure 2(b) shows a tree-like KG with a branching factor 3. We can see that the number of nodes at each level grows exponentially with their distance to the root of the tree. Due to this property, hyperbolic embeddings offer excellent quality for KG representation. As an example, we embed the toy KG into a 2-dimensional Poincaré disk. The hyperbolic embeddings enable all connected entities in the KG to be spaced equally far apart in 2-dimensional hyperbolic space and the hierarchical structure is preserved.

4 Methodology

We first associate each entity e with a vector as the feature. Specifically, we follow RNM (Zhu et al., 2021) to initialize the entity vector with the pre-trained semantic embedding \mathbf{x}^E , which can represent the lingual information of entity names. However, existing pre-trained word embedding is learned from Euclidean neural networks or in Euclidean space. To address this problem, we map Euclidean features into the hyperboloid manifold by:

$$\mathbf{x}^H = \exp_{\mathbf{o}}^c \mathbf{x}^E, \quad (4)$$

where $\mathbf{o} = \{0, 0, \dots, 0\} \in R^d$ is the original point. We consider \mathbf{x}^E as a vector in the tangent space

where \mathbf{o} is the reference point. The exponential map function $\exp_{\mathbf{v}}^c \mathbf{x}$ (Ganea et al., 2018) projects a tangent vector \mathbf{x}^E into the hyperbolic space at \mathbf{v} :

$$\exp_{\mathbf{v}}^c \mathbf{x} = \cosh(\sqrt{c}\|\mathbf{x}\|)\mathbf{v} + \frac{1}{\sqrt{c}} \frac{\sinh(\sqrt{c}\|\mathbf{x}\|)}{\|\mathbf{x}\|} \mathbf{x}. \quad (5)$$

To utilize the structure of the KGs for entity alignments, we introduce the aggregation operation in hyperbolic space. On each knowledge graph, we aggregate neighbor’s vectors with that of the center entity by:

$$\mathbf{z}_i^{(k+1)} = \mathbf{h}_i^{(k)} \oplus_c \mathbf{n}_i^{(k)}, k = 0, 1, \dots, K - 1 \quad (6)$$

where $\mathbf{h}_i^{(0)} = \mathbf{x}_i^H$ is the input feature and $\mathbf{n}_i^{(k)}$ is of aggregation of neighbor’s vectors according to their importance to the center entity. We use K to denote the depth. The existing attention mechanism utilizes the neural network layer to learn the weight as the importance of each neighbor. However, the linear function of the neural network layer is defined in the Euclidean space. To address this problem, we use the logarithmic function (Ganea et al., 2018) to map the hyperbolic vector into the tangent space at point \mathbf{u} :

$$\log_{\mathbf{u}}^c(\mathbf{y}) = \mathbf{d}_c(\mathbf{u}, \mathbf{y}) \frac{\mathbf{y} + \mathbf{c}\langle \mathbf{u}, \mathbf{y} \rangle_{\mathcal{M}} \mathbf{u}}{\|\mathbf{y} + \mathbf{c}\langle \mathbf{u}, \mathbf{y} \rangle_{\mathcal{M}} \mathbf{u}\|}. \quad (7)$$

Then we aggregate the neighbor’s vector in the tangent space and map them back to hyperbolic space:

$$\mathbf{n}_i^{(k)} = \exp_{\mathbf{h}_i^{(k)}}^c \left(\sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \log_{\mathbf{h}_i^{(k)}}^c(\mathbf{h}_j^{(k)}) \right), \quad (8)$$

where $\mathcal{N}(i)$ contains all neighbors of entity i and we consider the KG as an undirected graph. The importance $\alpha_{i,j}$ of neighbor j is learned from:

$$\alpha_{i,j} = \text{Softmax}_{j \in \mathcal{N}(i)}(\mathbf{Q}^{(k)} \cdot \text{CONC}((\log_{\mathbf{o}}^c \mathbf{h}_i^{(k)}), \log_{\mathbf{o}}^c \mathbf{h}_j^{(k)})) + \mathbf{q}^{(k)}. \quad (9)$$

Here we use $\text{CONC}(\cdot, \cdot)$ to denote the concatenation operation, and the Softmax function is used to normalize the weights. The attention mechanism enhances the important neighbors.

To further reduce the dimension of the hyperbolic vectors, we feed $\mathbf{z}_i^{(k+1)}$ into a hyperbolic linear layer:

$$\mathbf{h}_i^{(k+1)} = \text{exp}_o^c(\mathbf{W}^{(k)} \log_o^c(\mathbf{z}_i^{(k+1)})) \oplus_c \mathbf{b}^{(k)}, \quad (10)$$

where both $\mathbf{W}^{(k)}$ and $\mathbf{b}^{(k)}$ are learnable parameters. By iteratively executing Equations (6) and (10) for K times, we get a low dimensional vector in hyperbolic space for each entity. We remark that both Equations (6) and (10) are crucial to entity matching. Equation (6) can embed the structure information of KG and allows us to learn the smooth hyperbolic embeddings. Equation (10) can further learn information from the hidden state and reduce the dimension of the hyperbolic embeddings, which enables us to represent rich information with low dimensional vectors.

Find Equivalent Entities. Note that we have described how to learn hyperbolic embeddings for a single KG. Different from optimizing the embeddings of entities by only considering a single KG, we match two KGs in the same hyperbolic space to fine-tune the embeddings and learn a mapping function to find the equivalent pairs of entities.

Existing works either utilize a unified linear transformation function for all entities or directly enforce two embeddings close to each other. However, these works ignore the asymmetry of structure on different KGs and the localized nature of relations. Intuitively, the relations between two entities may vary from one pair to another. For example, the equivalent entity of "New Orleans" is "La Nouvelle-Orléans" in the French Wikipedia KG. However, the corresponding entity "Times Square" in French is still "Times Square". These two relations are totally different. The first relation is translation but the second pair of entities are identical to each other. As another example, there are 30,291 inner-relations between 15,000 entities in DBpedia KG. However, there are only 26,638 inner-relations between these entities in Yago KG. As a result, the structure of one DBpedia sub-graph may be same as its counterpart in Yago KG, where the structure of another sub-graph may be totally different from its counterpart. The asymmetry of relations on different KGs makes it difficult to learn a unified relationship for all pairs.

In this paper, we propose to use a localized mapping function. Specifically, for each entity, we learn the parameter in the mapping function from its hyperbolic embedding. Since KGs often exhibit hierarchies, the root node or the node at a low level on the KG tends to be located near the original point in the hyperbolic space generally. Considering this property, we design a parameterized geometric mapping function where the parameters are computed from the hyperbolic embedding.

Let \mathbf{H}^1 and \mathbf{H}^2 be the embeddings of entities after K iterations on G_1 and G_2 , respectively. Suppose we have a pair of equivalent entities across G_1 and G_2 : (e_i, e_j) where $e_i \in E_1$ and $e_j \in E_2$. Our instance-specific mapping function is a rotation:

$$f(\mathbf{H}_i^1) = \text{Rot}(\theta_i) \mathbf{H}_i^1 \quad (11)$$

where \mathbf{H}_i^1 is the hyperbolic embedding of entity e_i on G_1 and $\text{Rot}(\theta_i)$ is a block-diagonal matrix specified by 2×2 matrices commonly used in numerical linear algebra:

$$\begin{aligned} \text{Rot}(\theta_i) &= \text{diag}(G(\theta_{i,1}), \dots, G(\theta_{i,\frac{d}{2}})) \\ &= \begin{bmatrix} \cos(\theta_{i,1}) & -\sin(\theta_{i,1}) & & & & & & & & \\ \sin(\theta_{i,1}) & \cos(\theta_{i,1}) & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & & \sin(\theta_{i,\frac{d}{2}}) & \cos(\theta_{i,\frac{d}{2}}) & & & & \\ & & & & \sin(\theta_{i,\frac{d}{2}}) & \cos(\theta_{i,\frac{d}{2}}) & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & & & \end{bmatrix} \end{aligned} \quad (12)$$

The dimension d is an even number. We use the \mathbf{H}_i^1 to calculate $\theta_i = (\theta_{i,1}, \theta_{i,2}, \dots, \theta_{i,\frac{d}{2}})$:

$$\theta_i = \text{exp}_o^c(\mathbf{W}' \log_o^c(\mathbf{H}_i^1)) \oplus_c \mathbf{b}' \in R^{d/2}, \quad (13)$$

which makes the mapping adaptive to the entity.

To train the model, we propose to minimize the hyperbolic distance between $f(\mathbf{H}_i^1)$ and \mathbf{H}_j^2 for each pair of $S = \{(e_i, e_j) | e_i \in E_1, e_j \in E_2\}$. At the same time, we propose to push negative samples away from an entity. In order to achieve this, we design a loss function formulated as:

$$\begin{aligned} \text{loss} &= \sum_{(e_i, e_j) \in S} d_c(f(\mathbf{H}_i^1), \mathbf{H}_j^2) \\ &- \sum_{(e_{i'}, e_{j'}) \in S^-} d_c(f(\mathbf{H}_{i'}^1), \mathbf{H}_{j'}^2) + \gamma, \end{aligned} \quad (14)$$

where $\gamma > 0$ is a margin hyper-parameter and S^- represents the set of negative samples. We follow (Wu et al., 2019) to generate negative samples S^- .

Alignment Inference Strategy. Now we have the mapping function from G_1 to G_2 . For each entity $e_i \in E_1$, we find the aligned entity $\tilde{e}_j \in E_2$ by:

$$\tilde{e}_j = \underset{\tilde{e}_j \in E_2}{\text{argmin}} d_c(f(\mathbf{H}_i^1), \mathbf{H}_j^2). \quad (15)$$

5 Experiment

5.1 Experimental Setup

Dataset. We choose three Benchmark datasets in the experiment: EN-FR, EN-DE and D-Y (Sun et al., 2020c). Specifically, EN-DE represents two cross-lingual (English–German) KGs of DBpedia, where each KG contains 15K entities. Likewise, EN-FR contains 15K matches between English DBpedia and French DBpedia. D-Y maps 15K entities of DBpedia KG to 15K entities of Yago KG. We follow previous work (Sun et al., 2020c) to split all entity pairs into 20%/10%/70% for training, validation and test sets.

Baselines. We compare our approach against 12 state-of-the-art entity alignment methods. These baselines can be classified into three categories: (1) triple-based (MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), BootEA (Sun et al., 2018), RSNs (Guo et al., 2019) and MultiKE (Zhang et al., 2019)), (2) attributes-based (JAPE (Sun et al., 2017) and AttrE (Trisedya et al., 2019)) and (3) graph-based (GCNAlign (Wang et al., 2018), RDGCN (Wu et al., 2019), AliNet (Sun et al., 2020b), RNM (Zhu et al., 2021) and HyperKA (Sun et al., 2020a)). For all baselines, we use the default parameters as described in the corresponding paper.

Model Variants. To demonstrate the effectiveness of different components of our model, we implement three variants of our Geometric method for Entity Alignment in Hyperbolic space (GEA-H), including (1) **Xavier-I**: a variant of GEA-H to initialize the entity vectors with Xavier normal initializer instead of pre-trained semantic embeddings; (2) **Linear-T**: replaces our geometric method with a linear transformation (Sun et al., 2020a). Note that we use a hyperbolic neural network layer for transformation; (3) **Unified-R**: a rotation function with unified parameters for all entities instead of learning from the hyperbolic embeddings.

Performance Metrics. In our experiments, we use three widely used performance metrics: Hit@1, Hit@5 and MRR (Sun et al., 2020b; Wu et al., 2019; Zhang et al., 2019). Given an entity in one KG, we sort the list of entities in another KG according to the hyperbolic distance to the queried entity in ascending order. Then Hit@ k counts the proportion of entities in the test set whose aligned entity is in the top k list; while MRR averages the reciprocal ranks of the aligned entity in the sorted list. All reported performance results in the experi-

ment were averaged over 3 runs.

Model Configuration. We configure the negative constant curvature of hyperbolic space as a trainable parameter. We use a two-layer hyperbolic graph neural network, where the dimensions of hidden representations and output are 200 and 100 respectively by default. For the input layer, we initialize the entity vectors x with the pre-trained word embeddings (300-d) from the FastText model. If the entity name is null or not in the pre-trained dictionary, we use a random vector as initialization. For all baselines, we use the default parameters as described in the corresponding paper. In each epoch of the training process, we sample 125 negative pairs. We train our models using a Riemannian Adam optimizer with a learning rate of 0.001 and a weight decay of 0.01. All experiments are repeated 3 times and the average performance metrics over the 3 runs are reported to combat randomness.

5.2 Result

Quantitative Evaluation. Table 1 compares the alignment performance of the various approaches on three datasets, where the best results are shown in bold. We can see our full-fledged GEA-H consistently achieves the best performance on all three datasets, showing the advantages of GEA-H over entity alignment methods in Euclidean space. Specifically, our model gives 3% improvement in Hit@1 over the best baseline on EN-DE and D-Y. The performance on EN-FR slightly decreases for all methods. However, our GEA-H still outperforms these baselines. We can also observe 20% Hit@1, 10% Hit@5 and 0.2 MRR improvements over HyperKA (the only baseline in hyperbolic space) on average.

Several reasons lead to the advantage of our GEA-H over baselines. First, compared with these approaches based on embedding in Euclidean space, the hyperbolic embeddings can reserve hierarchical structures in KG with low dimension, which is vital for entity alignment. For example, an entity at low levels (e.g., "movie") is an unlikely equivalent to an entity (e.g., "Emma Stone"- the actress of La La Land) at high levels. Another important advantage of our GEA-H is that we use a geometric mapping method based on rotation instead of a linear transformation. Since the circle circumference and disc area grow exponentially with respect to the radius, the linear transformation can not address the non-linear nature of distance ratio in hyperbolic space. Our rotation method is designed

Table 1: Overall performance comparison

Model		EN-DE			EN-FR			D-Y		
		Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR	Hit@1	Hit@5	MRR
Triple-based	MTransE	0.307	0.518	0.407	0.247	0.467	0.351	0.463	0.675	0.559
	IPTransE	0.350	0.515	0.430	0.169	0.320	0.243	0.313	0.456	0.378
	BootEA	0.675	0.820	0.740	0.507	0.718	0.603	0.739	0.849	0.788
	RSNs	0.587	0.752	0.662	0.393	0.595	0.487	0.514	0.655	0.580
	MultiKE	0.756	0.809	0.782	0.749	0.819	0.782	0.903	0.939	0.920
Attributes-based	JAPE	0.288	0.512	0.394	0.262	0.497	0.372	0.469	0.687	0.567
	AttrE	0.517	0.687	0.597	0.481	0.671	0.569	0.668	0.803	0.731
Graph-based	GCNAlign	0.481	0.679	0.571	0.338	0.589	0.451	0.465	0.626	0.536
	RDGCN	0.830	0.895	0.859	0.755	0.854	0.800	0.931	0.969	0.949
	AliNet	0.615	0.771	0.684	0.387	0.613	0.487	0.591	0.722	0.650
	RNM	0.731	0.810	0.768	0.623	0.690	0.649	0.834	0.876	0.854
	HyperKA	0.622	0.827	0.713	0.403	0.660	0.519	0.614	0.806	0.699
Variants of GEA-H	Xavier-I	0.679	0.757	0.759	0.564	0.662	0.629	0.685	0.796	0.739
	Linear-T	0.674	0.767	0.718	0.539	0.645	0.589	0.654	0.734	0.693
	Unified-R	0.727	0.779	0.751	0.664	0.713	0.687	0.802	0.852	0.824
Full-fledged model	GEA-H	0.863	0.924	0.891	0.775	0.857	0.812	0.967	0.981	0.973

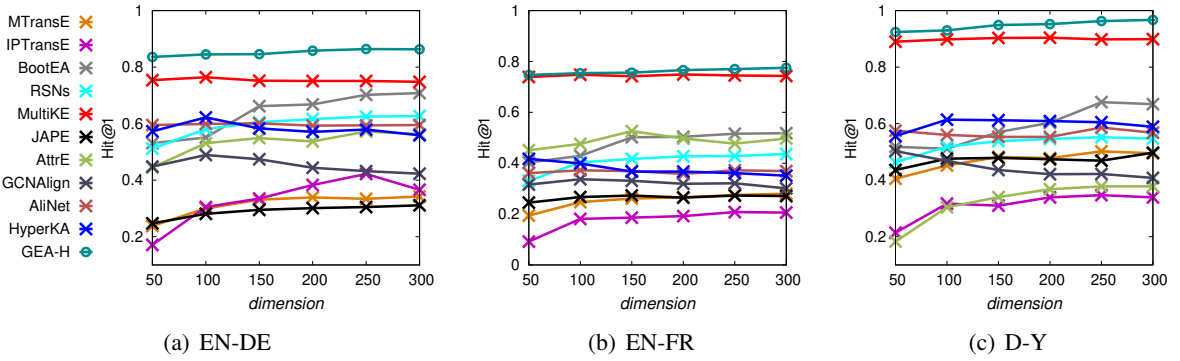


Figure 3: H@1 performance comparison using varying dimensions

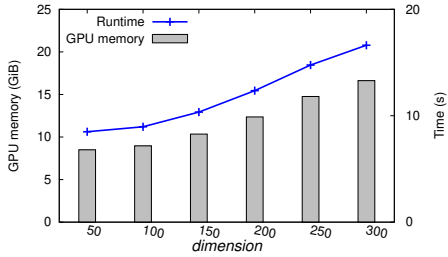


Figure 4: GPU memory cost and running time

to address this problem. Lastly, we learn instance-specific mapping parameters for each entity instead of using unified parameters. The localized parameters will address the locality of mapping.

5.3 Ablation Study

Initialization Method. Initialization can have a significant impact on neural network models. Compared with GEA-H, Xavier-I uses the Xavier normal initializer (Sun et al., 2020a) to initialize the entity vectors. Experimental results of GEA-H out-

perform the Xavier-I by a large margin consistently and it verifies the effectiveness of our initialization.

Effectiveness of Geometric Mapping. To verify the effectiveness of our rotation-based geometric mapping method, we replace Linear-T in HyperKA with a hyperbolic linear transformation for comparison. The experimental results in Table 1 show that our method outperforms Linear-T across all three datasets. This is because the linear transformation failed to address the non-linear distance ratio with respect to radius for nodes at different levels.

Localized Mapping We also investigate the effectiveness of our instance-specific mapping function. The variant Unified-R uses a unified mapping function for all entities instead of learning from entity embeddings. We compare Unified-R with our GEA-H across three datasets. The results indicate that our localized mapping function increases performance significantly and it is essential to learn the parameters adaptively.

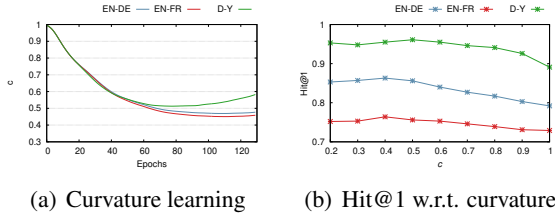


Figure 5: Effects of curvature

Table 2: Performance w.r.t. Negative Sampling Ratio

Ratio	EN-DE		EN-FR		D-Y	
	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
25	0.845	0.873	0.746	0.784	0.931	0.945
50	0.859	0.884	0.768	0.806	0.949	0.961
75	0.863	0.891	0.775	0.812	0.967	0.973
100	0.864	0.893	0.776	0.814	0.971	0.973
150	0.870	0.894	0.772	0.810	0.970	0.969

5.4 Sensitivity of Parameters

Effect of Varying Dimensions. The dimension of hyperbolic space plays a vital role in the expressiveness of our hyperbolic KG embeddings. To demonstrate the effectiveness of dimensions, we vary the length of hyperbolic embeddings from 50 to 300 with an interval of 50. Figure 3 shows Hit@1 on three datasets with varying dimensions. We also investigate the effectiveness of dimension for baselines whose dimension can be configured. Note that the dimension for RDGCN and RNM is not configurable (fixed at 300). As the dimension approaches 50, we can observe that the performances of some baselines decrease drastically on all three datasets. Our model offers much better representation and achieves the best performance in low-dimensional space. It validates our hypothesis that our method can solve the entity alignment problem in a low-dimensional hyperbolic space with a promising result. In addition, as shown in Figure 4, the occupied GPU memory and running time increase drastically as the dimension increases. Therefore, there is a trade-off between accuracy and computational cost.

Evaluation on number of negative instances. Note that we generate negative instances for training purposes. The performance of entity alignment is highly sensitive to the number of negative samples. In Table 2, we demonstrate the impact of sampling ratio (number of negative instances per pair) for GEA-H. It is clear that sampling more negative instances is beneficial for the performance of the model. On all three datasets, we observe limited performance improvement when the sampling ratio is beyond 75, which justifies our default parameter. The reason behind the performance return

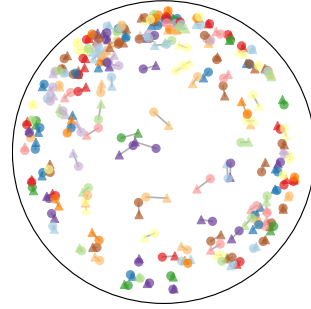


Figure 6: Visualization in 2-d space

is that there is trade-off between pushing negative samples away and minimizing the distance of positive samples. When there are too many negative samples, the loss of negative samples will have much more weight than positive samples. This will hurt the overall performance. Moreover, setting the sampling ratio too aggressively will only increase computation costs for training.

5.5 Effect of curvature

In hyperbolic space, the hierarchical structure can be reflected by the curvatures. With different values of curvature, the knowledge graph will be embedded into different hierarchical structures. In this paper, the value of curvature is trainable as a model parameter. Figure 5(a) shows the value of c in the training process. Note that the curvature of our model is initialized as 1 at the beginning. We can see that the value of c converges in the training process. To further investigate the effect of the curvature. We use a fixed curvature instead. As shown in Figure 5(b), the curvature learned by our model converges near the estimated optimal curvature.

5.6 Visualization

We visualize the 2-d embeddings (after mapping) for random pairs of entities from dataset "D-Y" in Poincaré disk. Figure 6 shows 200 pairs of entities in Poincaré disk, where entities on G_1 are marked with small circles and their corresponding entities on G_2 are marked with triangles. We can see that a pair of equivalent entities are closer to each other in the 2-d hyperbolic space.

6 Conclusion

We proposed GEA-H, a geometric entity alignment method in hyperbolic space. GEA-H learns low-dimensional hyperbolic embeddings for entities in KGs with an attention-based hyperbolic graph neural network. We design a geometric function based on rotation for mapping entities and learn the localized parameters of mapping function for each entity to address the locality of mapping.

Limitations

GEA-H is focused on an important task: matching equivalent entities across knowledge graphs and providing new tools to study KGs. We do not make any statements regarding its performance beyond this scope. One limitation of our work is that it requires a set of 1-to-1 alignments for training purposes. These alignments are supposed to be labeled manually.

References

- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 2787–2795.
- Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* 34, 4 (2017), 18–42. <https://doi.org/10.1109/MSP.2017.2693418>
- Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 1511–1517. <https://doi.org/10.24963/ijcai.2017/209>
- Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. *Proc. VLDB Endow.* 10, 5 (2017), 565–576. <https://doi.org/10.14778/3055540.3055549>
- Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic Neural Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 5350–5360.
- Congcong Ge, Xiaoze Liu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2021. Make It Easy: An Effective End-to-End Entity Alignment Framework. In *SIGIR 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 777–786.
- Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 855–864. <https://doi.org/10.1145/2939672.2939754>
- Hao Guo, Jiuyang Tang, Weixin Zeng, Xiang Zhao, and Li Liu. 2021. Multi-modal entity alignment in hyperbolic space. *Neurocomputing* 461 (2021), 598–607. <https://doi.org/10.1016/j.neucom.2021.03.132>
- Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to Exploit Long-term Relational Dependencies in Knowledge Graphs. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2505–2514. <http://proceedings.mlr.press/v97/guo19c.html>
- Richard A Harshman et al. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. (1970).
- Bo Hui and Wei-Shinn Ku. 2022. Low-rank Nonnegative Tensor Decomposition in Hyperbolic Space. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 646–654. <https://doi.org/10.1145/3534678.3539317>
- Bo Hui, Da Yan, Haiquan Chen, and Wei-Shinn Ku. 2022. Time-sensitive POI Recommendation by Tensor Completion with Side Information. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 205–217. <https://doi.org/10.1109/ICDE53745.2022.00020>
- Bo Hui, Da Yan, Wei-Shinn Ku, and Wenlu Wang. 2020. Predicting Economic Growth by Region Embedding: A Multigraph Convolutional Network Approach. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 555–564. <https://doi.org/10.1145/3340531.3411882>
- Viet-Phi Huynh and Paolo Papotti. 2019. Buckle: Evaluating Fact Checking Algorithms Built on Knowledge Bases. *Proc. VLDB Endow.* 12, 12 (2019), 1798–1801. <https://doi.org/10.14778/3352063.3352069>
- Chao Jiang, Yi He, Richard Chapman, and Hongyi Wu. 2022. Camouflaged Poisoning Attack on Graph Neu-

- ral Networks. In *ICMR '22: International Conference on Multimedia Retrieval, Newark, NJ, USA, June 27 - 30, 2022*, Vincent Oria, Maria Luisa Sapino, Shin'ichi Satoh, Brigitte Kerhervé, Wen-Huang Cheng, Ichiro Ide, and Vivek K. Singh (Eds.). ACM, 451–461. <https://doi.org/10.1145/3512527.3531373>
- Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. 2011. LogMap: Logic-Based and Scalable Ontology Matching. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 7031)*, Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist (Eds.). Springer, 273–288. https://doi.org/10.1007/978-3-642-25073-6_18
- Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=SJU4ayYgl>
- Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. SIGMa: simple greedy matching for aligning large knowledge bases. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthrusamy (Eds.). ACM, 572–580. <https://doi.org/10.1145/2487575.2487592>
- Bing Liu, Harrison Scells, Guido Zuccon, Wen Hua, and Genghong Zhao. 2021. ActiveEA: Active Learning for Neural Entity Alignment. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3364–3374. <https://doi.org/10.18653/v1/2021.emnlp-main.270>
- Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Boosting the Speed of Entity Alignment 10 ×: Dual Attention Matching Network with Normalized Hard Sample Mining. In *WWW 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 821–832.
- Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational Reflection Entity Alignment. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1095–1104. <https://doi.org/10.1145/3340531.3412001>
- Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 6338–6347.
- Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. 2018. Representation Tradeoffs for Hyperbolic Embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4457–4466. <http://proceedings.mlr.press/v80/sala18a.html>
- Zequan Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020a. Knowledge Association with Hyperbolic Knowledge Graph Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 5704–5716. <https://doi.org/10.18653/v1/2020.emnlp-main.460>
- Zequan Sun, Wei Hu, and Chengkai Li. 2017. Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10587)*, Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin (Eds.). Springer, 628–644. https://doi.org/10.1007/978-3-319-68288-4_37
- Zequan Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 4396–4402. <https://doi.org/10.24963/ijcai.2018/611>
- Zequan Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020b. Knowledge Graph Alignment Network with Gated Multi-Hop Neighborhood Aggregation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York,*

- NY, USA, February 7-12, 2020. AAAI Press, 222–229. <https://aaai.org/ojs/index.php/AAAI/article/view/5354>
- Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020c. A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs. *Proc. VLDB Endow.* 13, 11 (2020), 2326–2340. <http://www.vldb.org/pvldb/vol13/p2326-sun.pdf>
- Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity Alignment between Knowledge Graphs Using Attribute Embeddings. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 297–304. <https://doi.org/10.1609/aaai.v33i01.3301297>
- Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 349–357. <https://doi.org/10.18653/v1/d18-1032>
- Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5278–5284. <https://doi.org/10.24963/ijcai.2019/733>
- Chengjin Xu, Fenglong Su, and Jens Lehmann. 2021. Time-aware Graph Neural Network for Entity Alignment between Temporal Knowledge Graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 8999–9010. <https://doi.org/10.18653/v1/2021.emnlp-main.709>
- Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view Knowledge Graph Embedding for Entity Alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5429–5435. <https://doi.org/10.24963/ijcai.2019/754>
- Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian Suchanek. 2020. An Experimental Study of State-of-the-Art Entity Alignment Approaches. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1. <https://doi.org/10.1109/TKDE.2020.3018741>
- Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 4258–4264. <https://doi.org/10.24963/ijcai.2017/595>
- Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021. Relation-Aware Neighborhood Matching Model for Entity Alignment. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4749–4756. <https://ojs.aaai.org/index.php/AAAI/article/view/16606>