Global entity alignment with Gated Latent Space Neighborhood Aggregation

Wei Chen
School of Computer, Wuhan University of Technology
Wuhan, China
chenw720@foxmail.com

Xiaoying Chen*
School of Computer, Wuhan University of Technology
Hubei Credit Information Center
Wuhan, China
13971520707@139.com

Shengwu Xiong
School of Computer, Wuhan University of Technology
Wuhan, China
Sanya Science and Education Innovation Park of Wuhan University of Technology Sanya
xiongsw@whut.edu.cn

Abstract

Existing entity alignment models mainly use the topology structure of the original knowledge graph and have achieved promising performance. However, they are still challenged by the heterogeneous topological neighborhood structures, which could cause the models to produce different representations of counterpart entities. In the paper, we propose a global entity alignment model with gated latent space neighborhood aggregation (LatsEA) to address this challenge. Latent space neighborhood is formed by calculating the similarity between the entity embeddings, it can introduce long-range neighbors to expand the topological neighborhood and reconcile the heterogeneous neighborhood structures. Meanwhile, it uses vanilla GCN to aggregate the topological neighborhood and latent space neighborhood respectively. Then, it uses an average gating mechanism to aggregate topological neighborhood information and latent space neighborhood information of the central entity. In order to further consider the interdependence between entity alignment decisions, we propose a global entity alignment strategy, i.e., formulate entity alignment as the maximum bipartite matching problem, which is effectively solved by Hungarian algorithm. Our experiments with ablation studies on three real-world entity alignment datasets prove the effectiveness of the proposed model. Latent space neighborhood information and global entity alignment decisions both contributes to the entity alignment performance improvement.

1 Introduction

Knowledge graph (KG) is an important tool to store knowledge, which provides support for many applications, such as question answering systems, recommender systems. Many KGs have been constructed for particular applications. Sometimes single KG cannot meet the needs of certain applications, so it is necessary to integrate multiple complementary KGs. Entity alignment is non-trivial to integrating different KGs. Traditional entity alignment methods are mainly divided into two categories, one is based on the entity's label information (Ngomo and Auer, 2011; Pershina et al., 2015), the other is based on manually defining features. Recently, the embedding-based representation learning has attracted more and more attention. Given a KG, embedding-based representation learning can map entities to a low-dimensional vector space, then entity alignment model can find counterpart entities according to the similarity of the embeddings. The embedding-based entity alignment approaches show its superiority. Such methods are all built based on KG embedding models (such as TransE (Bordes et al., 2013)). Then they use a transformation matrix to convert the embedding space and complete the entity alignment by calculating the similarity between entity embeddings, such as MTransE (Chen et al., 2017). In recent researches, graph convolutional neural networks (GCN) have shown amazing results when processing graph structure data. GCN-based Entity alignment models all hope that counterpart entities have similar neighborhood structures and have similar vector representations. Several recent GCN-based alignment models (such as GCN-Align (Wang et al., 2018), AVR-GCN (Ye et al., 2019), GMNN (Xu et al., 2019), AliNet (Sun et al., 2020), RDGCN (Wu et al., 2019)) have achieved good results.

*corresponding author

However, existing embedding-based entity alignment models still face a critical problem. The counterpart entities usually have dissimilar neighborhood structures in between KGs, which will produce different embedding. The statistics on DBpedia datasets the percentages of such entity pairs reach 89.97% between Chinese-English, 86.19% between Japanese-English and 90.71% between French-English, respectively (Sun et al., 2020).

The difficulty in tackling this problem is how to alleviate the difference in topological neighborhood structure. Therefore, we propose a global entity alignment model with gated latent space neighborhood aggregation (LatsEA). Latent space neighborhood is formed in the embedding space by calculating the similarity between the entity embeddings, which can introduce long-distance neighbors to expand the topological neighborhood. The basic idea is to first learn KGs embeddings through KG embedding models and calculate the similarity between entity embeddings to generate latent space neighborhood. So, the model can be combined with many existing KG embedding models. Then, we use the GCN to aggregate topological neighborhood and latent space neighborhood respectively. Finally, we use an average gating mechanism to aggregate topological neighborhood information and latent space neighborhood information.

State-of-the-art models treat entities autonomously when determining entity alignment results. Intuitively, if a target entity is aligned to a source entity with higher confidence, it has a smaller chance to be aligned with another source entity. So, we formulate entity alignment as the maximum bipartite matching problem, which can be solved by Hungarian algorithm, so that the interdependence between entity alignment decisions can be captured to make global entity alignment decisions. Our experiments with ablation studies prove the effectiveness of the proposed model (LatsEA) and the global entity alignment strategy. In summary, our main contributions as follows:

- We propose a novel global entity alignment model, which introduces latent space neighborhood to reconcile the heterogeneous neighborhood structures.
- We propose an average gating mechanism to aggregate topological neighborhood information and latent space neighborhood information, so that the entity embeddings contain the entities information in topological and latent space neighborhood.
- We consider the interdependence between entity alignment decisions, and propose a global entity alignment strategy, i.e., formulate entity alignment as the maximum bipartite matching problem and use the classic Hungarian algorithm to solve.
- We perform experiments with ablation studies on three real-world datasets prove the effectiveness of the proposed model and the global entity alignment strategy.

2 Related Work

In our proposed entity alignment model, the most relevant works are KG embedding and embedding-based entity alignment, so we introduce some of the work related.

2.1 KG Embedding

In the past few years, the researches on KG embedding is mainly divided into three categories: translational distance models, neural network based models, and bilinear models (Zhang et al., 2020). TransE (Bordes et al., 2013) is a classic model in the translational distance models. It believes that in a triple, the relation is the translations of the head entities to the tail entities. TransE can solve the relation type of 1-1 well, but cannot solve the relation type of 1-N, N-1, N-N. Therefore, subsequent researches have proposed many improved versions based on the TransE, such as TransH (Wang et al., 2014), TransD (Lin et al., 2015).

With the rise of neural networks in recent years, neural networks are used to many KG embedding researches, such as ConvE (Dettmers et al., 2018) and CapsE (Nguyen et al., 2019). KGs have three widely spread types of relation patterns: symmetry, inversion, and composition (Chen et al., 2017). However, models mentioned above can model one or two type of relation patterns. RotatE (Sun et al.,

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

2019), which is a translational distance models, can model all relation patterns. It maps the entities and relations to the complex vector space and defines each relation as a rotation from the head entities to the tail entities. But real-world KGs usually have semantic hierarchies that RotatE fails to model. So, to tackle this challenge, the HAKE (Zhang et al., 2020) model is based on the RotatE, and retains the rotation characteristics of the RotatE. It contains modulus part and phase part. The modulus part models the entities at different levels of the hierarchy in KGs, and the phase part models the entities at the same level of the semantic hierarchy. In the paper, we use the HAKE (Zhang et al., 2020) model to learn KGs embedding and generate latent space neighborhood by calculating the cosine similarity between entity embeddings.

2.2 Embedding-based Entity Alignment

Embedding-based entity alignment models employ the embedding models to learn entity embeddings, then learn a mapping to transform the embedding space and find entity alignment through the similarity between these embeddings, such as MTransE (Chen et al., 2017). IPTransE (Zhu et al., 2017) iteratively adds the newly discovered entity alignment to the training data to improve the performance. There are also some models that use auxiliary information to improve the performance. For example, JAPE (Sun et al., 2017) combines attribute information to find entity alignment. Graph convolutional network (GCN) has demonstrated its powerful ability to process graph structure data, so many entity alignment models use GCN. GCN-Align (Wang et al., 2018) is a cross-language KG alignment framework, it uses GCN to convert two KGs into the same embedding space, and then calculates the distance between entity embeddings to completing the entity alignment. RDGCN (Wu et al., 2019) and AVR-GCN (Ye et al., 2019) improved the traditional GCN alignment framework. MuGNN (Cao et al., 2019) proposes a two-step method (KG completion and multi-channel graph neural network) for entity alignment. AliNet (Sun et al., 2020) introduces two-hop neighbors to reconcile heterogeneous neighborhood structures, and has achieved good results.

However, existing embedding-based entity alignment models mentioned above are still challenged that the neighborhood structure may be heterogeneous in different KGs. This may cause model to produce dissimilar entity embeddings for the corresponding entities, which will affect the accuracy of entity alignment. To tackle this problem, a straightforward idea is to mitigate the heterogeneity between KGs. AliNet (Sun et al., 2020) and MuGNN (Cao et al., 2019) above have noticed the problem. We are inspired by these works, to alleviate the heterogeneity, our idea is to utilize the latent space neighborhood in the KGs embedding space that can introduce distant neighbors to expand the topological neighborhood. In order to further consider the interdependence between entities that have been ignored in many existing researches, we convert the entity alignment into maximum bipartite matching problem to make global entity alignment. This problem can be further solved by the Hungarian algorithm. Our experiments with ablation studies on three entity alignment datasets prove the effectiveness of the proposed model. Latent space neighborhood information and global entity alignment decisions both contributes to the entity alignment performance improvement.

3 Problem Formulation

Formally, we represent a KG as a directed graph, where E represents the set of entities, R represents the set of relations, and T is the set of triples. Each triplet $(h, r, t) \in T$ indicates that the head entity h is connected to the tail entity t through the relation r. Entity alignment task is to automatically find more entity pairs that denote the same real-world identity in heterogeneous KGs. Entity alignment model is to take different KGs $G = (E, R, T)$ and $G' = (E', R', T')$ as input, and find a set of identical entities $S_e = \{(e, e') \in E \times E' \mid e \equiv e'\}$ representing the same object in the real-world. Those entity alignment seeds can be used as training data.

4 The Proposed Model

The goal of entity alignment is to discover the entity pairs representing the same object in heterogeneous KGs. In the paper, LatsEA uses HAKE (Zhang et al., 2020) model to learn the latent space of the KG,

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

then it aggregates the topological neighborhood and the latent space neighborhood of the central entity through vanilla GCN respectively. The average gating mechanism controls the topological neighborhood information and latent space neighborhood information aggregation to yield central entity embeddings. Then the entity similarity matrix is constructed by calculating the similarity between entity embeddings. To further consider the interdependence between entities and make global entity alignment decisions, we convert the entity similarity matrix into a bipartite graph, i.e., each row represents an entity in the source KG, and each column represents an entity in the target KG. So, the entity alignment problem is equivalent to maximum bipartite matching problem, which can be solved by Hungarian algorithm. The model architecture is illustrated in Figure.1.

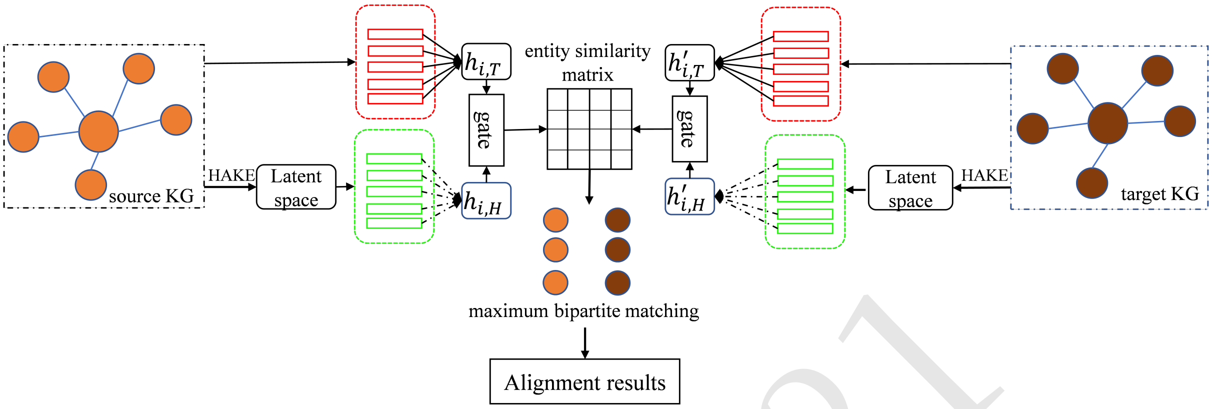

Figure 1: Overview of the LatsEA. The red rectangle in the figure represents the topological neighborhood, and the green rectangle represents the latent space neighborhood.

4.1 Basic Embedding Module

In our proposed model, in order to obtain a very meaningful entity embedding, the KG embedding model is required not only to model common relation patterns in the KG, but also to model semantic hierarchies. Because semantic hierarchies are very common in real-world applications. So, we choose Hierarchy-Aware Knowledge Graph Embedding (HAKE) (Zhang et al., 2020) to learn the latent space of KG. HAKE maps KG into the polar coordinate system, the modulus part can model entities at different level, and the phase part can model entities at the same level of the hierarchy. Given a triple (h, r, t), the formula of HAKE as follows:

$$\begin{aligned} h_m \circ r_m = t_m, \qquad where \quad h_m, t_m \in R^k, r_m \in R_+^k \\ (h_p + r_p) mod 2\pi = t_p, \qquad where \quad h_p, r_p, t_p \in [0, 2\pi)^k \end{aligned} \tag{1}$$

where h_m and h_p are generated by the modulus part and the phase part respectively. Therefore, the entity embeddings can be expressed as the concatenation of h_m and h_p. We calculate the cosine similarity between entity embeddings to generate the entity's latent space neighborhood, which will be described in next section. In practice, we train HAKE beforehand and freeze its parameters in the following process.

4.2 Topological Neighborhood and Latent Space Neighborhood

Existing entity alignment models mainly use the topology structure. But there is almost no way to use the structural information in latent space. The entity's topological neighborhood $N_T(u)$ is defined as follows:

$$N_T(u) = \{v | v \in E, (u, v) \in R\} \tag{2}$$

where E is a set of entities, R is a set of relations. With a certain number of hops, if an entity has a path connected to the central entity, it will be part of the topological neighborhood. As shown in Figure. 2, the set of all blue circles build up the topological neighborhood when the number of hops is 2.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

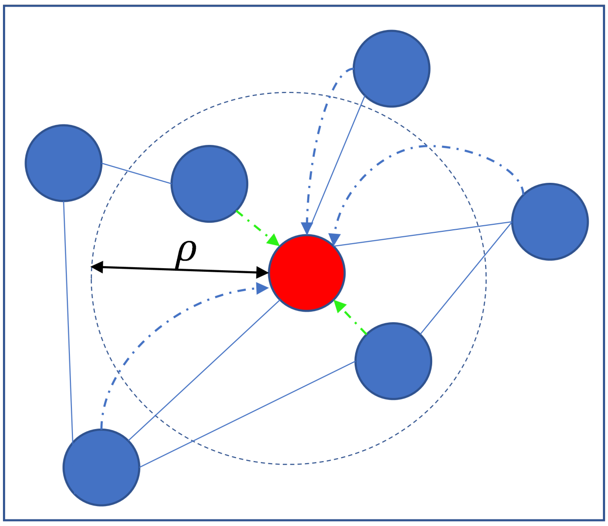

Figure 2: An illustration of the latent space neighborhood and topological neighborhood.

We introduce the latent space neighborhood to alleviate the heterogeneous. As shown in Figure. 2, if the distance between entity embeddings generated by HAKE is less than our predefined threshold ρ, the entity is the adjacent entity of the central entity in the latent space. These entities form the latent space neighborhood of the central entity. Therefore, the latent space neighborhood $N_H(u)$ can be expressed by the following formula:

$$N_H(u) = \{v|v \in E, d(e_u, e_v) < \rho\} \tag{3}$$

where ρ is a predefined distance threshold. e_u, e_v are embeddings of entity u and entity v, respectively. $d(e_u, e_v)$ is a function to calculate the distance between entity embeddings, we use the cosine similarity. The dotted circle in Figure. 2 represents the latent space neighborhood, the radius is the threshold ρ. The blue dotted line represents the topological neighborhood aggregation, and the green dotted line represents the latent space neighborhood aggregation.

4.3 Neighborhood Aggregation

The GCN was first proposed in ICLR (Kipf et al., 2017), which deals specifically with graph-structured data and particularly powerful. Therefore, we use vanilla GCN to aggregate the topological neighborhood and latent space neighborhood respectively.

Topological Neighborhood Aggregation. The one-hop neighborhood is the most important structure information to learn the central entities' embeddings. The hidden representation of entity i at l^{th} layer is represented as $h_{i,T}^{(l)}$, it is obtained by aggregating its one-hop neighborhood, the calculation method is as follows:

$$h_{i,T}^{(l)} = \sigma\Big(\sum_{j \in N_T(i)} \frac{1}{a_i} W_T^{(l)} h_j^{(l-1)}\Big) \tag{4}$$

Where $N_T(\cdot)$ is a set of one-hop neighbors (including itself) of entity i, a_i is the normalization constant, $W_T^{(l)}$ is the weight matrix of the l^{th} layer, $\sigma(\cdot)$ is an activation function, we use the hyperbolic tangent function (tanh). The input of each layer in GCN is a vertex feature matrix, it can encode nodes so that the vector representation of the entities contains the structural information in the graph.

Latent Space Neighborhood Aggregation. The latent space of the KG is learned through the HAKE (Zhang et al., 2020) embedding model, which maps the KG to a continuous latent space. The HAKE embedding model is equivalent to a mapping function, which maps the nodes into a vector. The neighborhood in the latent space contains the entity in the dotted circle with radius ρ in Figure. 2. The definition of the latent space neighborhood is described in section 4.2. It is a set of entities whose distance to the central entity is less than a given parameter ρ in the latent space. We use cosine similarity as the distance function $d(\cdot)$. Through the aggregation of latent space neighborhood, it is possible to capture long-range

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

dependencies and alleviate the non-isomorphic neighborhood structures. Vanilla GCN is also used to encode entities in latent space neighborhood aggregation. Like the topological neighborhood aggregation, the hidden representation of the entity i in the l^{th} layer is represented as $h_{i,H}^{(l)}$, which can be calculated using the following formula:

$$h_{i,H}^{(l)} = \sigma(\sum_{j \in N_H(i)} \frac{1}{a_i} W_H^{(l)} h_j^{(l-1)}) \tag{5}$$

Where $N_H(i)$ is a set of latent space neighbors (including itself) of entity i, a_i is the normalization constant, $W_H^{(l)}$ is the weight matrix of the l^{th} layer.

4.4 Gated Topological and Latent Space Neighborhood Aggregation

After generating topological neighborhood information $h_{i,T}^{(l)}$ and latent space neighborhood information $h_{i,H}^{(l)}$ through GCN aggregation, they need to be aggregated to generate central entities embeddings $h_i^{(l)}$ in the l^{th} layer. Because LSTM uses a gating mechanism to greatly alleviate the gradient disappearance and achieved relatively good results. Motivated by LSTM, we propose to use an average gating mechanism to aggregate topological neighborhood information and latent space neighborhood information. Different from AliNet (Sun et al., 2020), the average gating mechanism adds an average between different neighborhoods to reduce the influence of noise. First, the average vector $\bar{h}_i^{(l)}$ of topological neighborhood information $h_{i,T}^{(l)}$ and latent space neighborhood information $h_{i,H}^{(l)}$ is obtained at the l^{th} layer, its calculation method is as follows:

$$\bar{h}_i^{(l)} = \frac{1}{n} \sum_{j \in T,H} h_{i,j}^{l} \tag{6}$$

Where n is the number of neighborhood types. The basic idea is that if the entity embeddings in a neighborhood is far from the average embedding, vector averaging can alleviate the noise introduced in a neighborhood. Therefore, the hidden representation $h_i^{(l)}$ of entity i at l^{th} layer can be calculated by the following formula:

$$h_i^{(l)} = GM(\bar{h}_i^{(l)}) + (1 - GM(\bar{h}_i^{(l)})) \cdot h_{i,T}^{(l)} \tag{7}$$

$$GM(\bar{h}_i^{(l)}) = \sigma(W\bar{h}_i^{(l)}) \tag{8}$$

where $GM(\cdot)$ is a gate that controls the aggregation of topological neighborhood information and latent space neighborhood information, this gate is as shown in (8). W is a weight matrix, so that the gate can be trained together with the model. We can find from formula (7) that when the latent space neighborhood brings more noise, its weight will be reduced, which can well alleviate the influence of the noise introduced by the latent space neighborhood and improve the accuracy of the model.

4.5 Global Entity Alignment Strategy

We calculate the similarity between entity embeddings generated by average gated topological and latent space neighborhood aggregation to generate the similarity matrix M. Many state-of-the-art models have determined entity alignment results in an independent manner. However, each entity alignment decision is interdependent. Source KG and target KG have a vertex set respectively, entity alignment is equivalent to finding as many matching entities as possible in two disjoint vertex sets. So, to make global entity alignment decisions, we formulate entity alignment as the maximum bipartite matching problem, which can be solved by Hungarian algorithm. To facilitate subsequent processing, we use 1 to subtract the similarity matrix M to get the matrix M^-. Therefore, if the element in M^- is smaller, the corresponding two entities are more similar. The complete process is shown in Algorithm 1, which has time complexity of $O(n^3)$ for entity alignment.

Suppose there are three entities s_1, s_2, s_3 in the source KG, and three entities t_1, t_2, t_3 in the target KG. The example in Figure. 3 illustrates the algorithm.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

Algorithm 1: Hungarian algorithm

```
    Input: M⁻
    Subtracting the row minimum from each row;
    Subtracting the col minimum from each row;
    lineCount = 0;
    while (lineCount <len(M⁻)):
        Cover all zeros with a minimum line_count;
        if (lineCount == len(M⁻)):
            break;
        else:
            all uncovered elements in M⁻ subtract min uncovered element;
            all elements in M⁻ that are covered twice add min uncovered element;
    Find an optimal assignment in M⁻ with zeros cover;
```

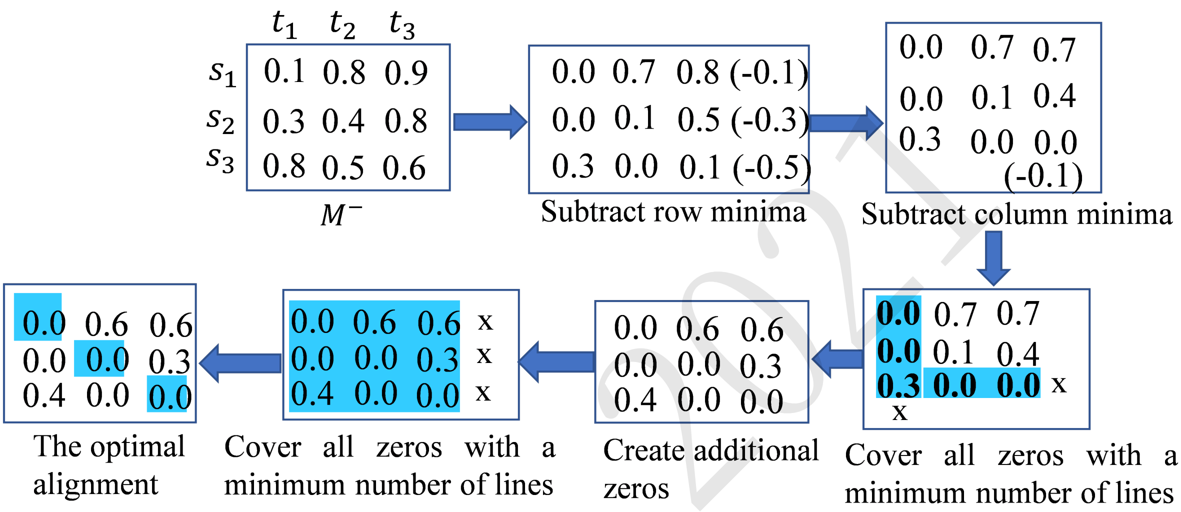

Figure 3: Entity alignment as maximum bipartite matching problem using Hungarian algorithm.

To enable LatsEA to let the embeddings of aligned entities have small distance while those of unaligned entities have large distance, we use a set of pre-aligned entity pairs S_e as training data to train LatsEA. We minimizing the margin-based ranking loss function L in (Wu et al., 2019) to train model's parameters in an end-to-end way.

$$L = \sum_{(u,v)\in S_e} \sum_{(u',v')\in S_e'} [f(h_u, h_v) + \gamma - f(h_{u'}, h_{v'})]_+ \tag{9}$$

Where $[x]_+ = max\{0, x\}$, S_e is the set of pre-aligned entity pairs, which is the positive sample during training. $S_e^{'}$ is the set of negative entity pairs yielded by corrupting (u, v). γ is margin hyper-parameters separating positive and negative entity pairs. $f(\cdot) = \|\cdot\|_2$ is the L2 distance metric function. We adopt Adam to minimize the loss function L. Then use the global entity alignment strategy to obtain the final entity alignment result.

5 Experiments

5.1 Datasets and Experiment Settings

Our experiments were conducted on DBP15K created by Sun et al. (2017). This dataset was generated based on the DBpedia multilingual KG and contains DBP_{ZH-EN} (Chinese-English), DBP_{JA-EN}

Methods	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
*MTransE	0.308	0.614	0.364	0.279	0.575	0.349	0.244	0.556	0.335
*JAPE	0.412	0.745	0.490	0.363	0.658	0.476	0.324	0.667	0.430
*AlignE	0.472	0.729	0.581	0.448	0.789	0.563	0.481	0.824	0.599
*GCN	0.487	0.790	0.559	0.507	0.805	0.618	0.508	0.808	0.628
MuGCN	0.493	**0.845**	0.611	0.501	**0.856**	0.620	0.498	**0.869**	0.622
AliNet	0.505	0.726	0.589	0.528	0.747	0.612	0.526	0.785	0.622
LatsEA	**0.522**	0.763	**0.613**	**0.539**	0.772	**0.625**	**0.538**	0.787	**0.632**

Table 1: Results comparison on entity alignment (* marks the results obtained from AliNet)

(Japanese-English) and DBP_{FR-EN} (French-English). Each dataset contains 15,000 aligned entities for training and testing models. For detailed statistics, please refer to the original paper.

We investigate LatsEA's performance on entity alignment by comparing with three embedding-based entity alignment models and three GCN-based entity alignment models. They are MTransE (Chen et al., 2017), JAPE (Sun et al., 2017), AlignE (Sun et al., 2018), GCN (Kipf et al., 2017), MuGCN (Cao et al., 2019) and AliNet (w/o rel. loss) (Sun et al., 2020). Because of the AliNet (Sun et al., 2020) model only provides codes that does not optimize the relation loss, we only compare with AliNet (w/o rel. Loss) (Sun et al., 2020). If LatsEA optimizes the relation loss proposed by AliNet, it will also perform better. We use grid search to find the appropriate value of the hyper-parameters. We choose hyper-parameters on the following possible values: margin γ in {0.5,1.0,...,3.0}, learning rate in {0.0001,0.001,0.01,0.1}, ρ in {0.80,0.81,...,0.95}, the dimension of the hidden representation of each layer in {200,300,400,500,600}, the number of GCN layers in {1,2,3,4}. The following hyper-parameters were used in the experiments.The γ is 1.5, the learning rate is 0.001, ρ is 0.93, the batch size is 1024. For each positive sample, 10 negative samples are collected. All GCN-based models stack two layers of GCN, and the dimension of the hidden representation of the three layers is 500, 400, and 300. We use $tanh(\cdot)$ as the activation function of neighborhood aggregation, and use the $ReLU(\cdot)$ as the activation function of gating mechanism. We use 70% of the seed alignments (10500 entity pairs for dbp15k) as the validation set and base on Hits@1 Performance termination training with a patience of 5 epochs. Following convention, we use Hits@1, Hits@10, and MRR to evaluate how the models perform. Higher Hits@1, Hits@10, and MRR scores indicate better performance.

5.2 Entity Alignment Results

Table 1 shows the performance of LatsEA and other entity alignment models. To show the fairness of the comparison, we reproduce the experimental results of MuGCN (Cao et al., 2020) and AliNet (w/o rel. Loss) (Sun et al., 2020) models under the same conditions. According to the experimental results, we found that LatsEA significantly outperforms the other entity alignment models on Hits@1 and MRR on the three datasets, the best results are highlighted in bold. For example, LatsEA achieves a Hits@1 score of 0.522, 0.539, and 0.538 on the three datasets, which are 0.017, 0.011, and 0.012 higher than the Hits@1 scores of AliNet (w/o rel. Loss) (Sun et al., 2020), respectively. Although LatsEA did not perform as well as MuGCN (Cao et al., 2020) on Hits@10, note that, the Hits@1 can better reflect the performance of the models. Because Hits@1 is equivalent to precision. LatsEA will perform better with global entity alignment strategy, and its results will be discussed in detail in section 5.5. We think that these results have demonstrated LatsEA's effectiveness. The reason is that LatsEA uses the latent space neighborhood information to mitigate heterogeneous neighborhood structures.

5.3 Effectiveness of Latent Space Neighborhood

Here, we set up two variants of LatsEA, LatsEA (To) only uses topological neighborhood information, LatsEA (Ls) only uses latent space neighborhood information. The entity alignment results are shown in Table 2. The best results are highlighted in bold. We observed that the latent space neighborhood brought improvement to entity alignment. This is because latent space neighborhood can introduce long-range

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

Methods	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
LatsEA(Ls)	0.115	0.372	0.197	0.113	0.257	0.166	0.105	0.269	0.162
LatsEA(To)	0.504	0.728	0.578	0.528	0.757	0.616	0.535	**0.794**	0.632
LatsEA	**0.522**	**0.763**	**0.613**	**0.539**	**0.772**	**0.625**	**0.538**	0.787	**0.632**

Table 2: Results on DBP15K w.r.t. different neighborhood

Methods	$DBP15K_{ZH-EN}$		
	H@1	H@10	MRR
*GCN	0.487	**0.790**	0.559
LatsEA(ρ=0.92)	0.516	0.742	0.601
LatsEA	**0.522**	0.763	**0.613**
LatsEA(ρ=0.94)	0.519	0.717	0.606
LatsEA(ρ=0.95)	0.517	0.714	0.604

Table 3: Results on $DBP15K_{ZH-EN}$ w.r.t. different distance

neighbors to mitigate heterogeneous neighborhood structures. For example, on $DBP15K_{ZH-EN}$, the introduction of latent space neighborhood information can improve the entity alignment performance by about 2% on Hits@1, about 4% on Hits@10 and MRR. When LatsEA uses the latent space neighborhood information alone, the model performs poorly. This may be because the threshold ρ is large, which results in less useful information contained in the generated latent space neighborhood.

5.4 Impact of Distance when Generating Latent Space Neighborhood

The purpose of introducing the latent space is to expand the neighborhood structure and reconcile the heterogeneity between KGs. We use grid search to find that the model performs better when $\rho = 0.93$. In order to deeply observe the influence of distance threshold ρ when generating latent space neighborhood, we evaluated the performance of LatsEA when the ρ is slightly higher than 0.93 or slightly lower than 0.93. The entity alignment results are shown in Table 3. Due to limited space, we only show the entity alignment results on the $DBP15K_{ZH-EN}$. The best results are highlighted in bold. It can be observed from the Table 3 that LatsEA performs best when the cosine similarity $\rho = 0.93$. We observe that the model's performance will declines as well when the ρ is larger. Because the larger ρ, latent space neighborhood will bring less useful information. Similarly, although the smaller ρ allows LatsEA to indirectly capture the more distant neighborhood information, such distant neighbors will also introduce more noise. Experiments results are consistent with our empirical analysis.

5.5 Effectiveness of Global Entity Alignment Strategy

Here, we examined the effectiveness of the proposed global entity alignment strategy. After generate the entity embeddings through our LatsEA, we obtain the similarity matrix of entities by calculating the similarities between these embeddings. In order to consider the interdependence between each entity alignment decision, we propose a global entity alignment strategy (GEAS). Concretely, we formulate entity alignment as the maximum bipartite matching problem, which can be solved by Hungarian algorithm. Because the algorithm can make global optimal entity alignment decisions and ensure that each entity gets the optimal match. Therefore, we use Hits@1 to evaluate whether our proposed GEAS is effective. Hit@1 is equivalent to the accuracy of the model. The accuracy is shown in Table 4. We can observe from Table 4 that the performance on all datasets will be better with GEAS, revealing the effectiveness of considering the interdependence between entity alignment decisions.

6 Conclusion

In this paper, we propose LatsEA for entity alignment, aiming at mitigating the heterogeneous among the topological neighborhood structures of counterpart entities. LatsEA introduces neighborhood struc-

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.

Methods	$DBP15K_{ZH-EN}$	$DBP15K_{JA-EN}$	$DBP15K_{FR-EN}$
	Hits@1	Hits@1	Hits@1
LatsEA	0.522	0.539	0.538
LatsEA+GEAS	**0.551**	**0.582**	**0.593**

Table 4: Results on DBP15K w.r.t. GEAS

ture in the latent space to capture distant neighbors' information, which can expand the topological neighborhood. It uses vanilla GCN to aggregate the information of adjacent entities in the topological neighborhood and latent space neighborhood, respectively. Then, it uses an average gating mechanism to aggregate topological neighborhood and latent space neighborhood information. In order to further consider the interdependence between entities and make global entity alignment decisions, we propose a global entity alignment strategy, i.e., formulate entity alignment as the maximum bipartite matching problem which is effectively solved by Hungarian algorithm. We evaluate LatsEA and global entity alignment strategy on three real-world entity alignment datasets, this result shows their effectiveness. Because the contribution of distant neighbors to the central entity embeddings is different, in future work, we will introduce an attention mechanism to give different weights to neighbor entities with different contributions when aggregating latent space neighborhood information.

Acknowledgements

This work was in part supported by the Major project of IoV , Technological Innovation Projects in Hubei Province (Grant No. ZDZX2020000027, 2019AAA024) and Sanya Science and Education Innovation Park of Wuhan University of Technology (Grant No. 2020KF0054).

References

Axel-Cyrille N Ngomo and Soren Auer. 2011. *LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data*, pp:1585–1590. IJCAI.

Maria Pershina, Mohamed Yakout and Kaushik Chakrabarti. 2015. *Holistic entity matching across knowledge graphs*, pp:1585–1590. International Conference on Big Data.

Antoine Bordes, Nicolas Usunier and Alberto Garcia-Duran. 2013. *Translating Em-beddings for Modeling Multi-relational Data*, pp:2787–2795. NIPS.

Muhao Chen, Yingtao Tian, Mohan Yang and Carlo Zaniolo. 2017. *Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment*, pp:1511–1517. IJCAI.

Zhichun Wang, Qingsong Lv, Xiaohan Lan and Yu Zhang. 2018. *Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks*, pp:349–357. EMNLP.

Rui Ye, Xin Li, Yujie Fang, Hongyu Zang and Mingzhong Wang. 2019. *A vectorized relational graph convolutional network for multi-Relational network alignment*, pp:4135—4141. IJCAI.

Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang and Dong Yu. 2019. *Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network*, pp:3156–3161. ACL.

Zequn Sun,Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang and Yuzhong Qu. 2020. *Knowledge Graph Alignment Network with Gated Multi-hop Neighborhood Aggregation*, pp:222–229. AAAI.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan and Dongyan Zhao. 2019. *Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs*, pp:5278–5284. IJCAI.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang and Jie Wang. 2020. *Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction*, pp:3065–3072. AAAI.

Zhen Wang, Jianwen Zhang, Jianlin Feng and Zheng Chen. 2014. *Knowledge Graph Embedding by Translating on Hyperplanes*, pp:1112–1119. AAAI.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu and Xuan Zhu. 2015. *Learning Entity and Relation Embeddings for Knowledge Graph Completion*, pp:2181–2187. AAAI.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp and Sebastian Riedel. 2018. *Convolutional 2D Knowledge Graph Embeddings*, pp:1811–1818. AAAI.

Dai Q. Nguyen, Thanh Vu, Tu D. Nguyen, Dat Q. Nguyen and Dinh Q. Phung. 2019. *A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization*, pp:2180–2189. NAACL-HLT.

Zhiqing Sun, Zhihong Deng, Jianyun Nie and Jian Tang. 2019. *RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space*. ICLR.

Hao Zhu, Ruobing Xie, Zhiyuan Liu and Maosong Sun. 2017. *Iterative Entity Alignment via Joint Knowledge Embeddings*, pp:4258–4264. IJCAI.

Zequn Sun, Wei Hu and Chengkai Li. 2017. *Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding*, pp:628–644. International Semantic Web Conference.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li and Tat S. Chua. 2019. *Multi-Channel Graph Neural Network for Entity Alignment*, pp:1452–1461. ACL.

Thomas N. Kipf and Max Welling. 2017. *Semi-Supervised Classification with Graph Convolutional Networks*. ICLR.

Zequn Sun, Wei Hu, Qingheng Zhang and Yuzhong Qu. 2018. *Bootstrapping Entity Alignment with Knowledge Graph Embedding*, pp:4396–4402. IJCAI.

Proceedings of the 20th China National Conference on Computational Linguistics, pages 1143-1153, Hohhot, China, Augest 13 - 15, 2021.