sensors

MDPI

Article

Enhancing Cross-Lingual Entity Alignment in Knowledge Graphs through Structure Similarity Rearrangement

Guiyang Liu [1,2], Canghong Jin [1,*], Longxiang Shi [1], Cheng Yang [1], Jiangbing Shuai [3] and Jing Ying [2]

[1] School of Computer and Computing Science, Hangzhou City University, Hangzhou 310015, China; zju_liu@zju.edu.cn (G.L.); shilx@hzcu.edu.cn (L.S.)
[2] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[3] Zhejiang Academy of Science & Technology for Inspection & Quarantine, Hangzhou 310051, China; sjb@zaiq.org.cn
* Correspondence: jinch@hzcu.edu.cn

Abstract: Cross-lingual entity alignment in knowledge graphs is a crucial task in knowledge fusion. This task involves learning low-dimensional embeddings for nodes in different knowledge graphs and identifying equivalent entities across them by measuring the distances between their representation vectors. Existing alignment models use neural network modules and the nearest neighbors algorithm to find suitable entity pairs. However, these models often ignore the importance of local structural features of entities during the alignment stage, which may lead to reduced matching accuracy. Specifically, nodes that are poorly represented may not benefit from their surrounding context. In this article, we propose a novel alignment model called SSR, which leverages the node embedding algorithm in graphs to select candidate entities and then rearranges them by local structural similarity in the source and target knowledge graphs. Our approach improves the performance of existing approaches and is compatible with them. We demonstrate the effectiveness of our approach on the DBP15k dataset, showing that it outperforms existing methods while requiring less time.

Keywords: knowledge graph; cross-lingual entity alignment; structural similarity rearrangement

Citation: Liu, G.; Jin, C.; Shi, L.; Yang, C.; Shuai, J.; Ying, J. Enhancing Cross-Lingual Entity Alignment in Knowledge Graphs through Structure Similarity Rearrangement. *Sensors* **2023**, *23*, 7096. https://doi.org/10.3390/s23167096

Academic Editor: Anabela Oliveira

Received: 26 June 2023
Revised: 22 July 2023
Accepted: 7 August 2023
Published: 10 August 2023

1. Introduction

Recently, knowledge graphs (KGs) have gained widespread adoption in AI-related fields, such as decision systems, knowledge reasoning, and recommendation systems, due to their excellent performance in storing structured data [1]. Knowledge graphs commonly consist of knowledge represented in the form of a triple, denoted by (h, r, t), in which h and t are entities, and there is a relation r between them. Although effective in representing structured data, problems caused by the underlying symbolic nature of such triples such as a lack of explicit semantics, absence of complex operations, and expressivity limitations usually make KGs hard to manipulate. To address these issues, research efforts have focused on the distributed representation of KGs, also known as KG embedding. KG embedding models embed a KG into a low-dimensional vector space by representing its entities and relations as semantic information vectors. The earliest work based on translation, TransE [2], considers a relation as the translation from its head entity to its tail entity. TransE can learn embeddings for all the elements of a ground truth triple (h, r, t) by assuming $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ are the embeddings of h, r, and t, respectively. Subsequent works have focused on enhancing TransE, such as TransD [3], TransH [4], and PTransE [5].

However, constructing KGs usually involves independent organizations or individuals, resulting in incomplete KGs that may not meet the needs of some applications. The above methods mainly model a single KG, but to obtain more comprehensive knowledge, it is necessary to integrate different knowledge graphs. Entity alignment [6] plays a critical role in knowledge fusion tasks. Entity alignment models fuse KGs by aligning

pairs of entities representing the same concept in the real world from different KGs, which may be in different languages. Cross-lingual entity alignment [7] integrates two or more KGs from other languages via pre-aligned seeds. Traditional entity alignment techniques focus on two perspectives: one is based on the equivalence reasoning specified by OWL semantics [8], and the other is based on similarity computation, which compares the symbolic features of entities [9,10]. Some studies have also focused on improving the accuracy of entity alignment through machine learning [11,12] and crowdsourcing [13].

Recent works devote a lot of effort to machine learning and deep learning approaches for knowledge graph entity alignment. MTransE [14] embeds KGs from different languages into different vector space, and learns a matrix for spatial transformation between monolingual vector spaces. JAPE [15] jointly embeds both the structural and attribute information of knowledge graphs into a unified vector space. BootEA [16] proposes a bootstrapping approach to iteratively generate new aligned entities as training data. In particular, deep learning methods have shown promising results in this area. GCN-Align [17] trains GCNs with pre-aligned seeds to embed entities of each language into a unified vector space. HGCN-JE [6] jointly learns entity and relation representations. RDGCN [18] incorporates relation information via attentive interactions between the knowledge graph and its dual relation counterpart to capture neighboring structures and learn better entity representations.

We find that all the methods mentioned above suffer from a limitation in that they only consider the globally closest entity in the vector space as an equivalent entity, which can be seen as a global optimum. However, when considering a specific node, the entity selected based on distance alone may not be the optimal cross-lingual counterpart, as it does not take into account the structural similarity between nodes. We can draw inspiration from SEU [19], which transformed cross-lingual knowledge graph entity alignment into an assignment task, as the original knowledge graph and the target knowledge graph share a similar graph structure, i.e., their adjacency matrices are similar. In this paper, we propose a cross-lingual entity alignment method based on local structural rearrangement. Our method calculates a k-nearest neighbor candidate set for each entity of the original knowledge graph during the alignment phase of the model. Subsequently, we rearrange each of the obtained candidate sets based on local structure, consisting of the surrounding nodes, to enhance the matching accuracy of equivalent entities.

As illustrated in Figure 1, the source knowledge graph (Chinese) contains relations between entities such as **"Ping Guo"**(N_{s_1}) and **"Qiao Bu Si"**(N_{s_2}), **"Jia Zhou"**(N_{s_3}) and **"Wei Ruan"**(N_{s_5}). For instance, **"Qiao Bu Si"** founded **"Ping Guo"** and **"Ping Guo"** is located in **"Jia Zhou"**. When searching for the cross-lingual correspondence of entity **"Ping Guo"** in the target knowledge graph (English), there are two possible results: **"Apple"**(N_{t_a}) and **"Apple"**(N_{t_b}). In this case, the local structure of entities must be taken into consideration. As the cross-lingual equivalent entities include **"Steven Jobs"**(N_{t_2}) in the target KG and **"Qiao Bu Si"** in the source KG, **"California"**(N_{t_3}) in the target KG and **"Jia Zhou"** in the source KG, and **"Microsoft"**(N_{t_5}) in the target KG and **"Wei Ruan"** in the source KG, entity **"Apple"**(N_{t_a}) has a similar structure to entity **"Ping Guo"**. Therefore, entity **"Apple"**(N_{t_a}) is more likely to be the cross-lingual counterpart of **"Ping Guo"** than the other entity **"Apple"**(N_{t_b}). In summary, we summarize the three significant contributions as follows:

- We introduce an approach called structure similarity rearrangement (SSR) to improve the precision of cross-lingual knowledge graph entity alignment, which centers around the formulation of a joint evaluation function that incorporates the local structural similarity of entities.
- We perform extensive experiments on the publicly available datasets DBP15k ZH_EN, JA_EN, and FR_EN, and demonstrate that our proposed model outperforms other state-of-the-art methods in the evaluation metrics of Hits@1.
- By conducting comparative experiments, we have discovered that our proposed method exhibits a seamless integration capability with other alignment models, leading to a substantial improvement over techniques (such as GCNs) that solely focus on

node-level features. Moreover, our model demonstrates considerable accuracy gains when compared to approaches (e.g., RDGCN) that consider the surrounding context.

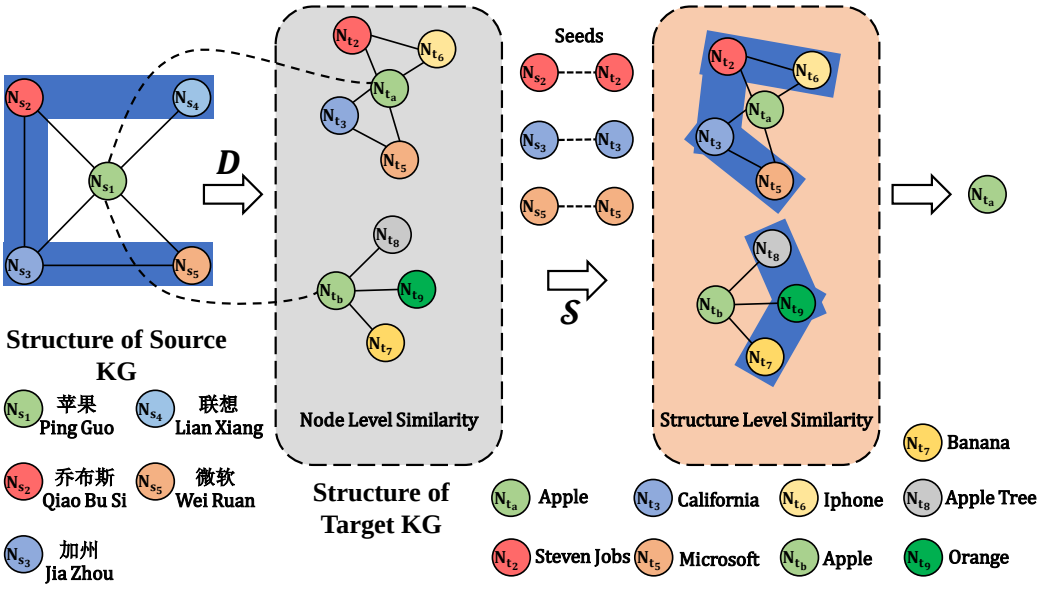


Figure 1. An example of cross-lingual KG entity alignment.

2. Related Work

The task of knowledge graph entity alignment can be broken down into two primary stages: (i) knowledge graph representation, and (ii) entity alignment based on the entity-level representation. In recent years, knowledge graph embedding techniques have been widely employed for distributed representation of the structural information in knowledge graphs.

2.1. Knowledge Graph Embedding

Knowledge graph (KG) is a technique that uses graph models to describe knowledge and model the association relationships between things [20–22]. KGs are composed of triples, $\langle entity, relation, entity \rangle$, and entities that have attribute–value pairs, which are connected by relationships to form a web-like structure [23,24]. However, manipulating KGs can be challenging due to the symbolic nature of triples. To address this issue, knowledge graph embedding (KGE) [25] has emerged as a promising research direction. KGE aims to map KG components into a continuous low-dimensional vector space that preserves the original structure of the KG and simplifies operations. Various embedding methods have been proposed, including TransE [2], which interprets a relation as the translation vector from the head entity to the tail entity in embedding space; TransH [4], which models a relation as a vector on a specific relationship hyperplane and learns different representations for an entity; and TransD [3], which uses two vectors to represent the semantics of an entity (relation) and construct a mapping matrix dynamically. Recently, a GCN-based approach called CompGCN [26] has been proposed, which leverages a range of entity–relation composition operations from KG embedding techniques to jointly embed both nodes and relations in a multi-relational graph.

2.2. Entity Alignment

Entity alignment is a crucial task in knowledge graph (KG) research which aims to identify entities that correspond to the same real-world object across different KGs. In the past, manual feature extraction or crowdsourcing [10,27–29] were often used to accomplish this task, which required substantial manual involvement. Recently, KG embedding techniques have been widely employed to facilitate entity alignment. JE [30] proposed

to jointly learn embeddings of multiple KGs in a uniform vector space to align entities in different KGs. MTransE [14] embeds two KGs into different vector spaces and aligns them by learning two transforming matrices. GCN-Align [31] is the first to employ graph convolutional networks (GCNs) to encode entities and attributes into a unified space for entity alignment.

3. Problem Formulation

3.1. Knowledge Graph

A knowledge graph KG is composed of sets of entities E, relations R, and triples T, where $T \subseteq E \times R \times E$ and $KG = (E, R, T)$.

3.2. Knowledge Graph Entity Alignment

An entity alignment model aims to automatically identify all of the corresponding entities of two given KGs. Without loss of generality, we choose one KG as the source knowledge graph and the other as the target knowledge graph, and denote them as $KG_s = (E_s, R_s, T_s)$ and $KG_t = (E_t, R_t, T_t)$.

Candidate set and ranking indicator. In the process of entity alignment, the model endeavors to discover a fixed number of cross-lingual counterparts in KG_t for each entity node e_i originating from KG_s. Subsequently, these counterparts are sorted in ascending order based on vector distance, yielding the candidate set $CAND(e_i)$ specific to the respective entity node. For any node e_j in the candidate set of e_i from KG_t, its ranking indicator based on vector distance can be defined as the difference between the length of the candidate set ($cand$) and its position in the candidate set ($index(CAND(e_i), e_j)$).

Similarity of local structures. Let G represent a graph, which can be denoted as $G = \langle V, E \rangle$, where V and E correspond to the sets of nodes and edges in the graph, respectively. We refer to the graph structure of KG_s as G_s, and similarly, the graph structure of KG_t as G_t. The local structure of a vertex is defined by the edges directly adjacent to it and the vertices connected to it via those edges. Thus, for a given node v_s from G_s and a node v_t from G_t, we employ the symbol $\mathcal{S}$ to represent the degree of similarity between their respective local structures.

Problem Statement Given two knowledge graphs KG_s and KG_t, the crux of the problem is to compute and rearrange the candidate set associated with each entity originating from KG_s.

The symbols used in our method are summarized in Table 1.

Table 1. Important notation.

Symbol	Definition
$KG = (E, R, T)$	A KG consisting of sets of entities E, relations R, and triples T.
E_s, E_t	The sets of entities from KG_1 and KG_2.
e_i	An entity from KG.
vec	The entity embeddings.
$CAND(e_i)$	The candidate set of entity e_i.
$CAND(E_s)$	The candidate sets of E_s.
$CAND_{new}(e_i)$	The rearranged candidate set of entity e_i.
$CAND_{new}$	The rearranged candidate sets of E_s.
$cand$	The size of candidate set.
$rank(e_i)$	The ranking of entity e_i.
$G = \langle V, E \rangle$	A graph G consisting of set of nodes V and set of edges E.
$\mathcal{S}$	The similarity of local structures.
$index(list, e)$	The index of e in $list$.

4. The Proposed Approach

Most of the existing models for entity alignment in knowledge graphs are composed of two main modules: an embedding module and an alignment module. The former is

responsible for generating distributed representations of the input knowledge graphs, whereas the latter aims to identify similar cross-lingual counterparts in the vector space. However, the aforementioned methodologies primarily focus on devising intricate network models to acquire a highly precise distributed representation of the knowledge graph. Nevertheless, the alignment phase still adheres to the conventional approach of employing a simple greedy search. Taking inspiration from SEU [19], we acknowledge that when it comes to two knowledge graphs that pertain to the same domain, their overall graph structure exhibits similarities, and the structure surrounding equivalent entities is also comparable. In this work, we propose a structure-based approach for rearranging the candidate sets obtained by the alignment module. Specifically, given two knowledge graphs KG_s and KG_t in different languages, our model first generates their distributed representations via the embedding module. Next, the alignment module employs the KNN algorithm to search for cross-lingual counterparts in the target knowledge graph for each entity in the source knowledge graph, and generates the corresponding candidate sets. We assume that the graph structures around the alignment nodes in the source and target knowledge graphs are similar, such that entities with similar graph structures in the source knowledge graph are likely to have similar counterparts in the target knowledge graph.

In order to enhance the precision of cross-lingual entity alignment, we propose a novel approach to rearrange the candidate sets of nodes in consideration of the similarity between the graph structures of the nodes in the source KG and their candidate set in the target KG. This structure-based rearrangement aims to modify the distribution of the candidate sets, and, therefore, improve the overall accuracy of the alignment task.

4.1. Overall Architecture

As illustrated in Figure 2, we propose a straightforward approach to enhance the precision of cross-lingual entity alignment. Firstly, the embedding representation of the knowledge graph is generated by leveraging an embedding module, such as JAPE, GCNs, HGCN, or RDGCN. Subsequently, a KNN algorithm is typically employed as the alignment module to obtain the cross-lingual corresponding candidate set for each node in the source knowledge graph. To optimize this candidate set, a rearrangement algorithm is utilized. The detailed process of our proposed approach is elaborated in Algorithm 1.

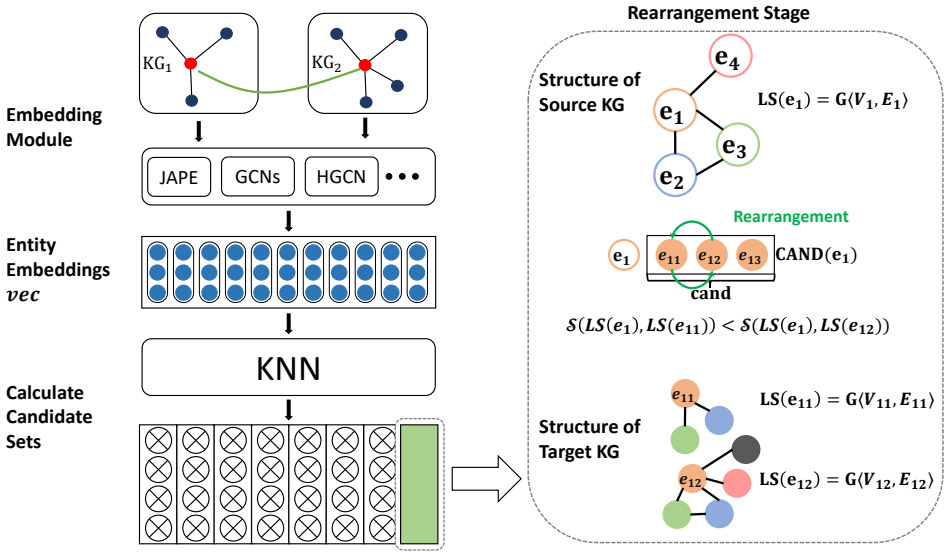


Figure 2. Overall architecture of the proposed approach.

Algorithm 1: Structural Enhancement Rearrangement Algorithm.

Input: Entity Embeddings *vec*, Test Data (E_s, E_t), Size of candidate set *cand*, $G_s = \langle V_s, E_s \rangle$, $G_t = \langle V_t, E_t \rangle$
Output: Rearranged Candidate Sets $CAND_{new}$, *Hits*@1.

calculate candidate sets $CAND(E_s)$;
for e_i *in* KG_s **do**
  $V_{e_i} \leftarrow \{e_j | e_j \in V_s \wedge (e_i, e_j) \in E_s\}$;
  $E_{e_i} \leftarrow \{(x, y) | x \in (V_{e_i} \cup \{e_i\}) \wedge y \in (V_{e_i} \cup \{e_i\}) \wedge (x, y) \in E_s\}$;
  $LS(e_i) \leftarrow G\langle V_{e_i}, E_{e_i} \rangle$;
  for v_j *in* $CAND(e_i)$ **do**
    $rank(v_j) \leftarrow cand - index(CAND(e_i), v_j)$;
    $V_{v_j} \leftarrow \{v_i | v_i \in V_t \wedge (v_i, v_j) \in E_t\}$;
    $E_{v_j} \leftarrow \{(x, y) | x \in (V_{v_j} \cup \{v_j\}) \wedge y \in (V_{v_j} \cup \{v_j\}) \wedge (x, y) \in E_t\}$;
    $LS(v_j) \leftarrow G\langle V_{v_j}, E_{v_j} \rangle$;
    if $LS(e_i) \sim LS(v_j)$ **then**
      $rank(v_j) \leftarrow rank(v_j) + \mathcal{S}$;
  $CAND_{new}(e_i) \leftarrow$ rearrange $CAND(e_i)$ by the updated *rank* of each candidate node;
$CAND_{new} \leftarrow CAND_{new} \cup CAND_{new}(e_i)$;
calculate *Hits*@1;

4.2. Embedding Module

The proposed approach primarily focuses on the structure-based rearrangement of candidate sets, and, thus, the choice of embedding module is flexible. For the purpose of illustration, we adopt the GCN alignment model as the embedding module. GCNs [17] are neural networks that can directly process graph-structured data of arbitrary shape and size. Specifically, a GCN layer receives both the node feature vector and the structural information of a graph. The structural information of the graph can be learned, and the feature vectors of all nodes can be updated through a GCN layer. By aggregating information from the neighboring nodes, GCNs update the feature vectors of nodes and are commonly employed in graph classification and regression tasks.

The input to a GCN model layer is the vertex feature matrix of the graph and the output is a new feature matrix obtained using the following equation:

$$H^{(l+1)} = \sigma\big(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\big). \tag{1}$$

where $H^{(l)} \in \mathbb{R}^{n \times d^{(l)}}$ is a matrix of $d^{(l)}$-dimensional feature vectors of n nodes in the l-th layer; σ is an activation function; $A \in \mathbb{R}^{n \times n}$ is the connectivity matrix of the graph; $\hat{A} = A + I$, and $I \in \mathbb{R}^{n \times n}$ is the identity matrix; $\hat{D} \in \mathbb{R}^{n \times n}$ is the diagonal node degree matrix of $\hat{A}$; and $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is the weight matrix of the l-th layer of the GCN model; the new vertex features have $d^{(l+1)}$ dimensions.

The GCN-Align model integrates the structural and attribute information of nodes to generate node embeddings, and since our method rearranges nodes in a candidate set based on structural similarity, the attribute information of nodes is not considered and only structural information is utilized for training.

4.3. Calculate Candidate Sets

A cross-lingual entity counterpart is predicted by computing the distance between the entity embeddings of two KGs in the vector space. For each pair of nodes, e_i in KG_1 and v_j in KG_2, the distance between them is computed using the following equation:

$$D(e_i, v_j) = ||\mathbf{e}_i - \mathbf{v}_j||_1. \tag{2}$$

where $\mathbf{e}_i$ and $\mathbf{v}_j$ denote the embeddings of e_i and v_j, respectively.

For an entity e_i in KG_1, our model computes the distance between e_i and all entities in KG_2 and returns a list as the candidate set in which the entities are sorted by distance from smallest to largest, because two entities with smaller distance are more likely to be equivalent. As a result, for n entity nodes in KG_1, a candidate sets matrix $M_c \in \mathbb{R}^{n \times l_c}$ can be obtained, where l_c denotes the length of each candidate set.

4.4. Local Structural Similarity

In the graph theory domain, the local structural analysis [32] of a node e_i in a graph $G = \langle V, E \rangle$ has been widely studied. Herein, V and E correspond to the sets of nodes and edges in the graph, respectively. Specifically, the local structure LS of the node e_i is formally defined as follows:

$$LS(e_i) = G\langle V_{e_i}, E_{e_i} \rangle. \tag{3}$$

where $V_{e_i} = \{e_j | e_j \in V \wedge (e_i, e_j) \in E\}$ and $E_{e_i} = \{(x, y) | x \in (V_{e_i} \cup \{e_i\}) \wedge y \in (V_{e_i} \cup \{e_i\}) \wedge (x, y) \in E\}$. The local structure of each node is a subgraph of the whole knowledge graph and there are a lot of methods [33] to determine whether two graphs are similar. We propose a simple but novel equation for computing graph similarity in this article, which is outlined as follows:

$$\mathcal{S}_{node}(e_i, v_j) = \sum_{x \in V_{e_i}} [\epsilon - \min_{y \in V_{v_j}} D(\mathbf{x}, \mathbf{y})]_+ \tag{4}$$

$$\mathcal{S}_{edge}(e_i, v_j) = \sum_{(x_i, y_i) \in E_{e_i}} |\{(x_j, y_j) | cond\}|. \tag{5}$$

$$\mathcal{S}(LS(e_i), LS(v_j)) = \lambda \times \mathcal{S}_{node}(e_i, v_j) + \delta \times \mathcal{S}_{edge}(e_i, v_j). \tag{6}$$

where *cond* in Equation (5) denotes $(x_j, y_j) \in E_{v_j} \wedge x_j \in CAND(x_i) \wedge y_j \in CAND(y_i)$; $\mathcal{S}_{node}(e_i, v_j)$ in Equation (4) and $\mathcal{S}_{edge}(e_i, v_j)$ in Equation (5) are similarities of neighboring nodes and edges, respectively, and λ and δ in Equation (6) are their weight coefficients.

In Equation (4), $[a]_+$ denotes the positive part of a, while $\epsilon > 0$ signifies the maximum admissible distance among equivalent entities, and its value can be specified as the maximum embedding distance observed within the training set for equivalent entities. $D(\mathbf{x}, \mathbf{y})$ denotes the embedding distance between entities x and y. The computation of $\mathcal{S}_{node}(e_i, v_j)$ is based on the concept that the similarity between e_i and v_j increases as the number of equivalent entities in V_{e_i} and V_{v_j} grows. To satisfy this condition, our approach involves identifying the node y in V_{v_j} with the closest embedding distance $\min_{y \in V_{v_j}} D$ to x. If the distance $\min_{y \in V_{v_j}} D(\mathbf{x}, \mathbf{y})$ is smaller than the threshold ϵ, we consider y as a potential equivalent node of x. In such cases, we accumulate the difference between ϵ and $\min_{y \in V_{v_j}} D(\mathbf{x}, \mathbf{y})$ into $\mathcal{S}_{node}(e_i, v_j)$. Conversely, if the embedding distance exceeds ϵ, we discard y as a potential equivalent node. In essence, the higher the count of potential equivalent nodes in V_{e_i} and V_{v_j}, and the smaller the embedding distance among these potential equivalents, the greater the theoretical validity of equivalent entities, resulting in a larger value of $\mathcal{S}_{node}(e_i, v_j)$.

In Equation (5), edge (x_j, y_j) satisfying the specified condition *cond* is regarded as analogous to (x_i, y_i). The term $\sum_{(x_i, y_i) \in E_{e_i}} |\{(x_j, y_j) | cond\}|$ computes the count of edges in E_{v_j} that meet the condition *cond*, and thereby quantifies their similarity to edges in

E_{e_i}. As the number of such similar edges in E_{e_i} and E_{v_j} increases, so does the value of $\mathcal{S}_{node}(e_i, v_j)$.

Equation (6) considers both node and edge similarities in the graph and can be utilized to obtain the similarity between two local structures. A higher value of $\mathcal{S}(LS(e_i), LS(v_j))$ indicates greater similarity between the two local structures.

4.5. Rearrangement Stage

This part is the most crucial component of our proposed approach. In this step, we rearrange the nodes in the candidate set matrix obtained in the previous step. The rearrangement process is depicted in Algorithm 1 and Figure 3, which takes as input the entity embedding matrix *vec* generated by the embedding module, the testing data (E_s, E_t), the candidate set size *cand*, and the adjacency matrix *adj* that encodes the structural information of the KGs. The algorithm outputs the rearranged candidate sets $CAND_{new}$ and the corresponding values of Hits@1.

In extant methodologies, the alignment module predominantly relies on the distance metric as the criterion for exploring cross-lingual correspondences. In this study, we present a novel metric called **Joint Similarity** as delineated below:

$$\mathcal{D}(e_i, v_j) = |D(e_i, v_j) - \mathcal{S}(LS(e_i), LS(v_j))|. \tag{7}$$

which stipulates that the proximity between two nodes ought to decrease concomitantly with the enhancement in the similarity of their corresponding local structures.

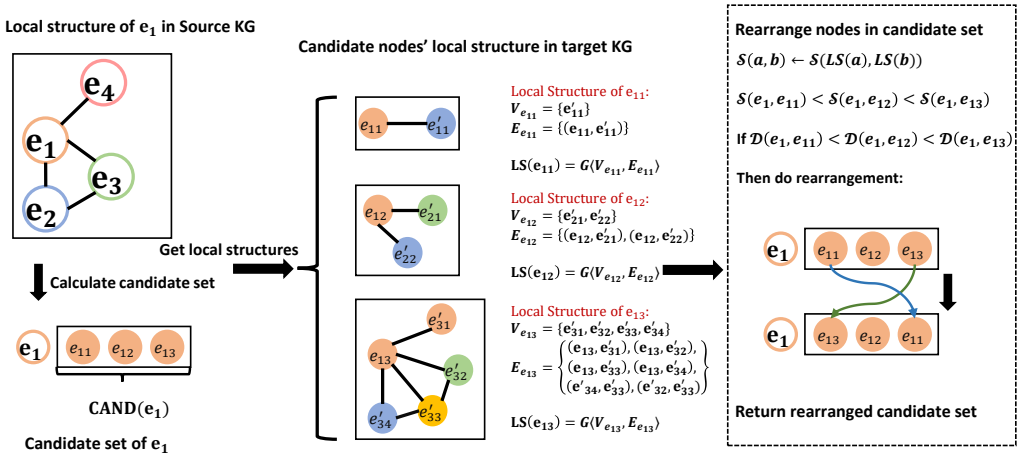


Figure 3. Entity rearrangement process. Firstly, we calculate the local structure of all nodes within the candidate set of e_1 via Equation (3). Secondly, we compute the local structural similarity between e_1 and the nodes within the candidate set using Equation (6). Following this, the joint similarities are determined through Equation (7). Lastly, to complete the rearrangement, the nodes within the candidate set are rearranged based on their respective values of joint similarity.

The details of the implementation of the algorithm are described as follows. Firstly, for each node e_i in the testing data from the source KG, we construct a map with the entity node's id of e_i as the key and its sum of ranking as the value and save its local structure into a sparse matrix. When traversing each node e_j in the candidate set of e_i, we record the ranking of e_j at first, and then add it with local structural similarity. Here, the ranking is defined as the size of the candidate set minus the subscript of the node in the candidate set. Then, we can obtain a map of the candidate set for e_i. By rearranging this map by value from largest to smallest, we obtain a new candidate set while considering local structural similarity. In theory, the ground truth cross-lingual correspondence in the new candidate set is ranked higher than in the original candidate set. Since not all nodes in the original candidate set satisfy local structural similarity, the length of the new candidate set

is reduced, making it difficult to compute values other than Hits@1. Our approach focuses on improving Hits@1, i.e., alignment accuracy improvement.

4.6. Complexity Analysis

Algorithm 1 presents the key methodology of our proposed algorithm. It consists of two nested loops and a conditional structure for judgment. The outer loop traverses every node e_i in the source knowledge graph KG_1, with a time complexity of $\mathcal{O}(n_{e_s})$, where n_{e_s} denotes the total number of entities in KG_1. Meanwhile, the inner loop iterates through each candidate node v_j in the candidate set of e_i, with a time complexity of $\mathcal{O}(cand)$. Within this loop, the algorithm acquires the local structure and computes the similarity of local structures with a time complexity of $\mathcal{O}(n_{r_t})$, where n_{r_t} represents the total number of edges in the target graph. Furthermore, if the local structure of e_i is similar to that of v_j, the algorithm proceeds to compute their similarity. In summary, the overall time complexity of the algorithm is $\mathcal{O}(cand \times n_{e_s} \times n_{r_t})$.

The adoption of a fixed value for *cand*, limited to the range of $[1, 100]$, serves to significantly reduce the time complexity of our proposed approach. Moreover, the alignment process exclusively utilizes nodes present in the testing data rather than traversing the entire knowledge graph. This selective strategy substantially reduces the overall count of nodes and edges, resulting in a significant decrease in the algorithm's time complexity.

5. Experiment

The proposed method is evaluated on the *DBP*15*k* datasets, aiming to address the following research questions:

- **RQ1**: To what extent can the integration of SSR with the baseline model's embedding module enhance the prediction accuracy of the entity alignment task?
- **RQ2**: What is the impact of varying the rearranging ranges on the overall performance of the model?
- **RQ3**: How do the parameters λ and δ in Equation (6) affect the performance of the model?

5.1. Datasets

In our experiments, we utilized the DBP15k datasets proposed by Sun et al. [15], (Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding in *International Semantic Web Conference*, Springer, Cham, 2017) This dataset comprises four language-specific KGs extracted from DBpedia for English (En), Chinese (Zh), French (Fr), and Japanese (Ja), with each KG containing approximately 65 k–106 k entities. Additionally, the dataset provides aligned entity sets for English and three other languages, with each set containing more than 15 k cross-lingual entity pairs. Statistical information about the datasets is shown in Table 2.

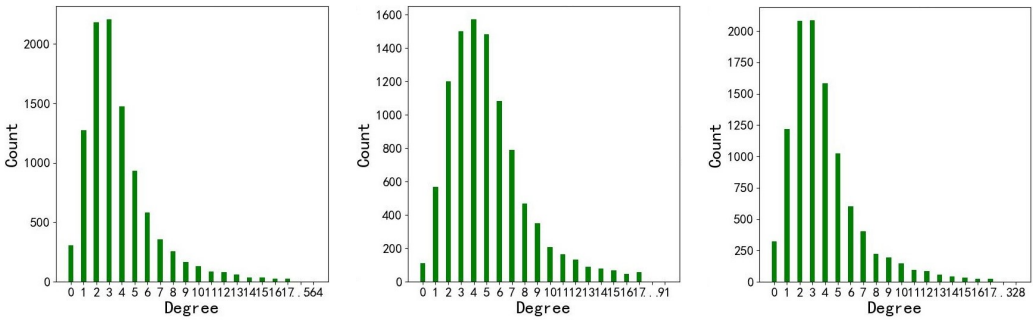
Table 2. Statistical information of DBP15k.

Dataset	Languages	Entities	Relations	Attributes	Rel. Triples	Attr. Triples
ZH_EN	Chinese	66,469	2830	8113	153,929	379,684
	English	98,125	2317	7173	237,674	567,755
JA_EN	Japanese	65,744	2043	5882	164,373	354,619
	English	95,680	2096	6066	233,319	497,230
FR_EN	French	66,857	1379	4547	192,191	528,665
	English	105,889	2209	6422	278,590	576,543

Structure Statistic

The degree of nodes is one of the most notable metrics that can provide insight into graph structure. Therefore, we conducted an analysis of the degree-related information

across the three language versions of the DBP15k dataset, which we present in Figure 4 and Table 3.



(**a**) (**b**) (**c**)

Figure 4. Degree distribution of local structure in DBP15k. (**a**) DBP15k$_{ZH_EN}$; (**b**) DBP15k$_{JA_EN}$; (**c**) DBP15k$_{FR_EN}$.

Table 3. Cardinality of local structure diversity.

Dataset	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
Number	81	80	108

The node degree distribution across the three language versions of the DBP15k dataset is presented in Figure 3. The horizontal axis indicates various degree of nodes, while the vertical axis represents the number of nodes with the corresponding degree. Our analysis reveals that the distribution trends of node degrees in the knowledge graphs of the three language versions are relatively consistent. The number of nodes tends to increase and then decrease with the rise in node degrees, and the majority of nodes exhibit degrees between 2 and 5.

In Table 3, we present the statistics of node degree types across the DBP15k training set in three distinct language versions. The dataset $DBP15k_{ZH\_EN}$ contains 81 distinct types of node degrees, and $DBP15k_{JA\_EN}$ features 80 distinct types of node degrees, exhibiting only marginal variation from the former. However, $DBP15k_{FR\_EN}$ displays 108 different types of node degrees, which is likely due to the greater linguistic similarity between French and English compared to Chinese or Japanese. This linguistic proximity results in a higher count of cross-lingual entity links between French and English in DBpedia. The discrepancies in the number of node degree types across the three datasets are likely to have a significant impact on the outcome of the subsequent experiments.

5.2. Experiment Settings

In light of the current state of entity alignment research, we observe that GCN-based methods have exhibited the highest performance on the DBP15k dataset and do not necessitate the incorporation of extra training data (https://paperswithcode.com/sota/entity-alignment-on-dbp15k-zh-en (accessed on 5 July 2023)). While PSR [34] and EMGCN [35] exhibit superior performance, they necessitate supplementary training data, which is unsuitable in our case. Therefore, we have chosen several representative GCN-based methods (without extra training data) as baseline techniques for our comparative evaluation. Additionally, we have included JAPE, a translation-based model that has achieved the highest performance among non-GCN-based methods, as a baseline for comparison.

During the experiments, we conducted a control test by selecting only the SE variants of each baseline approach since our rearrangement method does not require the use of attribute information in the knowledge graph.

The inter-language links in each dataset are divided according to the gold standard. In all control experimental groups, we use 70% of the inter-language links as the test data

and 30% of them as the train data. The main focus of the experiments is on the enhancement in Hits@1, i.e., the prediction accuracy of the corresponding entities across languages.

Apart from the hyperparameters that are included in our approach for the prior-order model, such as the division ratio between the training and validation sets and the learning rate, the length of the candidate set to be rearranged and the weight of neighbor are crucial hyperparameters. In our experiments, we test three candidate set lengths: 5, 10, and 50 to determine the optimal value, meanwhile we set the weights as $\lambda = 0.25$, $\delta = 0.5$.

5.3. Baselines Information

In our controlled experiments, we chose four primary baseline approaches, including a representative translation-based model, JAPE, and for GCN-based models, GCN-Align, HGCN-JE, RDGCN, and AliNet. As our approach is focused on local structural similarity and does not require the utilization of attribute-related information, we only compared our method with the SE variants of each baseline approach in our comparative experiments.

JAPE [15] is a joint attribute-preserving embedding model for cross-lingual entity alignment, which jointly embeds the structures of two KGs into a unified vector space and further refines it by leveraging attribute correlations in the KGs. In our experiment, we set its hyperparameters to the best-performing values: $d = 75, \alpha = 0.1, \beta = 0.05, \delta = 0.05$. The learning rate of SE is set to 0.01 based on empirical evaluation.

GCN-Align [31] is a graph convolutional network-based approach that leverages entity relations to construct the network structure of GCNs, enabling it to generate embeddings for each entity in the source and target KGs. For our experiments, we selected the hyperparameters that have shown the most promising results, including setting $d_s = 1000$ and $d_a = 100$ for the source and target entity embeddings, respectively. Additionally, we set the margin $\gamma_s = \gamma_a = 3$ in the loss function and empirically set β in the distance measure to 0.9.

HGCN-JE [6] is a joint entity–relation learning framework for entity alignment that minimizes human involvement and associated costs in seed alignment construction. For the experiment, we set the hyperparameters to $\gamma = 1$, $\beta = 20$, and the learning rate of 0.01. We also sample $\mathcal{K} = 125$ negative pairs every epoch.

RDGCN [18] is a novel relation-aware dual-graph convolutional network to incorporate relation information via attentive interactions between the knowledge graph and its dual relation counterpart, and further capture neighboring structures to learn better entity representations. In our experiments, we set $\beta_1 = 0.1$, $\beta_2 = 0.3$, and $\gamma = 1.0$ as the hyperparameters. The hidden representations in the dual and primal attention layers have dimensions of $d = 300$, $d' = 600$, and $\tilde{d} = 300$. The dimension of hidden representations in GCN layers is 300. The learning rate is set to 0.001, and we sample $\mathcal{K} = 125$ negative pairs every 10 epochs.

AliNet [36] introduces distant neighbors to expand the overlap between their neighborhood structures, which employs an attention mechanism to highlight helpful distant neighbors and reduce noise. Subsequently, it controls the aggregation of both direct and distant neighborhood information using a gating mechanism. In our experiments, we set the number of layers of AliNet to 2, the activation function for neighborhood aggregation to $\tanh()$, the one for the gating mechanism is $\mathrm{RelU}()$, and the dimensions of the three layers (including the input layer) to 500, 400, and 300, respectively. For each pre-aligned entity pair, we sample 10 negative samples.

5.4. Results

Tables 4–6 show the results of all experiments on the DBP15k dataset.

Table 4. Experiment results on DBP15k$_{ZH_EN}$.

Type	Model	Hits@1	Hits@5	Hits@10
NEM [1]	JAPE	39.39	62.86	71.02
	JAPE–SSR–5	**44.00**	-	-
	JAPE–SSR–10	43.91	-	-
	JAPE–SSR–50	40.65	-	-
NEM	GCN-Align	39.16	63.16	69.93
	GCNs–SSR–5	**44.26**	-	-
	GCNs–SSR–10	43.99	-	-
	GCNs–SSR–50	41.32	-	-
SEM [2]	AliNet	51.71	72.13	85.07
	AlN–SSR–5	**53.98**	-	-
	AlN–SSR–10	53.39	-	-
	AlN–SSR–50	49.76	-	-
SEM	HGCN-JE	69.31	80.92	84.70
	HJ–SSR–5	**71.02**	-	-
	HJ–SSR–10	68.30	-	-
	HJ–SSR–50	59.98	-	-
SEM	RDGCN	72.02	82.70	85.58
	RG–SSR–5	**72.66** *	-	-
	RG–SSR–10	67.26	-	-
	RG–SSR–50	62.90	-	-

[1] Abbreviation of node-embedding model. [2] Abbreviation of structure-embedding model.

Table 5. Experiment Results on DBP15k$_{JA_EN}$.

Type	Model	Hits@1	Hits@5	Hits@10
NEM	JAPE	33.84	57.39	67.03
	JAPE–SSR–5	**38.20**	-	-
	JAPE–SSR–10	38.15	-	-
	JAPE–SSR–50	35.53	-	-
NEM	GCN-Align	40.40	65.50	73.44
	GCNs–SSR–5	**44.90**	-	-
	GCNs–SSR–10	44.09	-	-
	GCNs–SSR–50	40.73	-	-
SEM	AliNet	51.70	72.88	79.62
	AlN–SSR–5	**53.19**	-	-
	AlN–SSR–10	52.30	-	-
	AlN–SSR–50	48.78	-	-
SEM	HGCN-JE	76.09	84.26	87.99
	HJ–SSR–5	**78.18**	-	-
	HJ–SSR–10	70.50	-	-
	HJ–SSR–50	61.93	-	-
SEM	RDGCN	77.84	88.30	90.79
	RG–SSR–5	**78.84** *	-	-
	RG–SSR–10	71.40	-	-
	RG–SSR–50	62.93	-	-

Table 6. Experiment Results on DBP15k$_{FR_EN}$.

Type	Model	Hits@1	Hits@5	Hits@10
NEM	JAPE	28.75	54.88	64.43
	JAPE–SSR–5	**33.56**	-	-
	JAPE–SSR–10	<u>33.45</u>	-	-
	JAPE–SSR–50	31.19	-	-
NEM	GCN-Align	40.11	66.51	75.98
	GCNs–SSR–5	**43.76**	-	-
	GCNs–SSR–10	<u>42.83</u>	-	-
	GCNs–SSR–50	38.60	-	-
SEM	AliNet	52.35	75.69	82.40
	AlN–SSR–5	**54.57**	-	-
	AlN–SSR–10	<u>53.36</u>	-	-
	AlN–SSR–50	48.71	-	-
SEM	HGCN-JE	<u>88.95</u>	93.12	94.75
	HJ–SSR–5	**89.67** *	-	-
	HJ–SSR–10	77.54	-	-
	HJ–SSR–50	64.29	-	-
SEM	RDGCN	<u>88.79</u>	94.57	95.91
	RG–SSR–5	**89.06**	-	-
	RG–SSR–10	76.25	-	-
	RG–SSR–50	64.40	-	-

The baseline methods exhibit distinct model characteristics, and can be categorized into two types: node-embedding models and structure-embedding models. Specifically, node-embedding models such as JAPE and GCN-Align solely consider node-level features, while structure-embedding models such as AliNet, HGCN-JE, and RDGCN learn not only node features, but also consider the surrounding environment information of the nodes.

In this study, we conducted a controlled experiment to evaluate the structural embedding variants (SEs) of all of the baseline methods. We computed their Hits@k values with k set to 1, 5, and 10. Subsequently, we employed the proposed SSR method to rearrange the corresponding candidate sets of cross-lingual entities obtained for each model with the corresponding model SSR variants. Here, XXX–*SSR*–c denotes the model utilizing an embedding module named XXX in combination with SSR, with the candidate set rearranging range set to $cand = c$.

As the SSR method only requires Hits@1 evaluation metrics, we report only the results for the Hits@1 column for the SSR variants. For the controlled experiments of each SSR variant with the original model, we emphasize the optimal Hits@1 results in **bold**, the suboptimal Hits@1 results underlined, and annotate the optimal results on the entire data set with * in the upper right corner of the values.

5.4.1. Rearranging Range

As mentioned earlier, an important parameter of the algorithm proposed in this paper is the length of the rearrangement range of the candidate set, and this parameter has an impact on the performance of the model, which is reflected in the experimental results of different SSR variants. From the experimental results, we can tentatively conclude that the prediction accuracy Hits@1 of the variant model decreases as the value of *cand* increases when the value of *cand* is taken in the range of three numbers $5, 10, 50$.

We apply the SSR method to the JAPE and GCN-Align models, and the experimental results show that the SSR variants of Hits@1 have a higher improvement relative to the original model when *cand* is taken as 5 and 10. Even when *cand* is taken to be 50, the SSR variants achieve a boost. This is due to the fact that both the JAPE and GCN-Align models are simpler node-embedding models, and the quality of the learned entity embeddings is

average. The local structure-based rearrangement is able to adjust the distribution of nodes in the candidate set, thus achieving higher prediction accuracy.

For the AliNet method, Hits@1 improves to different degrees when the *cand* of the variant is taken as 5 and 10, but the variant with $cand = 50$ does not perform as well as the original model. Furthermore, when we apply SSR to the HGCN-JE and RDGCN embedding modules, Hits@1 improves slightly only when *cand* takes 5. This is because the neural network designs of AliNet, HGCN-JE, and RDGCN are more complex. Among them, HGCN-JE introduces highway GCN, while RDGCN introduces relation-aware dual-graph. Although the training dataset of these two methods is limited to DBP15k, they can generate more data based on DBP15k due to the characteristics of the network structure, which is essentially equivalent to using exogenous data. AliNet introduces "distant neighbors" and an attention mechanism to consider the environmental information around the nodes, which also helps to improve the quality of the entity embedding vectors.

Through experimental observation, we have discovered a significant correlation between the lift rate of Hits@5 of the original model relative to Hits@1 of the original model, and the lift rate of Hits@1 of the SSR variant with $cand = 5$ relative to Hits@1 of the original model. We conducted an in-depth analysis and comparison of these relationships. The results are presented in Figure 5. The horizontal axis of each figure indicates the original lift rate, which is the lift rate of Hits@5 of the original model relative to Hits@1 of the original model. The vertical axis represents the model lift rate, which is the lift rate of Hits@1 of the SSR variant with $cand = 5$ relative to Hits@1 of the original model. Our analysis reveals a clear positive correlation between the two, where an increase in the original lift rate corresponds to an increase in the model lift rate.

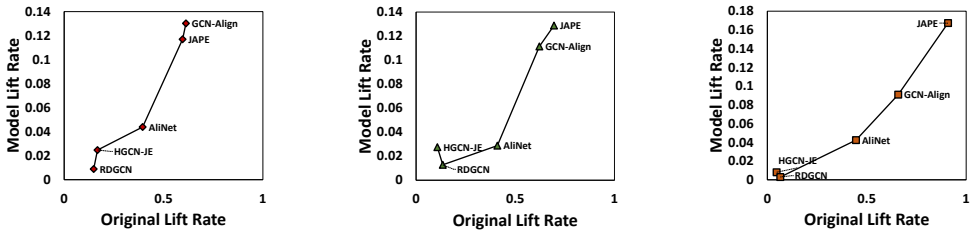


Figure 5. Models' performance on DBP15k. (**a**) DBP15k$_{ZH_EN}$; (**b**) DBP15k$_{JA_EN}$; (**c**) DBP15k$_{FR_EN}$.

5.4.2. SSR vs. JAPE

In our comparative experiments, we investigate the effectiveness of the structure-enhanced rearrangement method by applying it to JAPE and evaluating its impact on entity alignment accuracy across three different cross-lingual KG datasets: $DBP15k_{ZH\_EN}$, $DBP15k_{JA\_EN}$, and $DBP15k_{FR\_EN}$. Our results indicate that the performance improvement of JAPE varies across different datasets, with the largest improvement achieved on $DBP15k_{ZH\_EN}$, followed by $DBP15k_{JA\_EN}$, and the smallest improvement on $DBP15k_{FR\_EN}$. Specifically, the accuracy rate of Hits@1 can be increased by around 5% when *cand* is set to 5, and still improved by about 1% even when *cand* is increased to 50. Such improvements are statistically significant and demonstrate the effectiveness of our approach.

The structure embedding part of JAPE uses a traditional translation-based model, i.e., given a relational triple (h, r, t), we expect $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Such a simple model neglects the impact of the local structural information on the embeddings of entities, thereby presenting an opportunity to enhance JAPE's performance through our proposed structure-enhanced rearrangement approach.

5.4.3. SSR vs. GCNs

In the comparison experiments with GCNs, we observed that the original GCNs approach yielded almost identical Hits@1 results across three distinct datasets. On the other hand, our

proposed method showed improvements on all three datasets, with a 5% improvement on $DBP15k_{ZH\_EN}$, 4.5% improvement on $DBP15k_{JA\_EN}$, and 3.6% on the $DBP15k_{FR\_EN}$ dataset.

The GCNs method trains by optimizing the loss function globally for all nodes in the graph. Our method applies a structure-enhanced rearrangement to the candidate set, which takes into account the local structural similarity of the graph. Unlike the GCN model, which only considers the distance similarity between single nodes in the alignment module, our approach also considers the similarity of nodes and their surrounding structures when evaluating equivalent entities, resulting in higher final values of Hits@1.

5.4.4. SSR vs. AliNet

We conducted experiments on the DBP15k dataset using the AliNet model. Notably, our findings indicate that AliNet achieved comparable results across all three versions, without exhibiting the significant performance gains observed in the HGCN and RDGCN models on $DBP15k_{FR\_EN}$. Moreover, a careful examination of the experimental results reveals that the integration of our proposed approach into the AliNet model led to a noteworthy improvement of 2% to 3% on each of the three datasets.

5.4.5. SSR vs. HGCN

The performance of HGCN on the DBP15k benchmark dataset exhibits significant improvement compared to JAPE and GCNs. In particular, HGCN achieves the highest accuracy on the $DBP15k_{FR\_EN}$ dataset, where JAPE and GCNs show inferior performance.

As previously noted, there are more structural variations present in the $DBP15k_{FR\_EN}$ dataset compared to $DBP15k_{ZH\_EN}$ and $DBP15k_{JA\_EN}$. This may explain why HGCN achieves a higher accuracy rate on the $DBP15k_{FR\_EN}$ dataset.

Upon applying our proposed approach to the HGCN model, we observed an improvement of over 1% on both the $DBP15k_{ZH\_EN}$ and $DBP15k_{JA\_EN}$ datasets, but only a 0.7% improvement on $DBP15k_{FR\_EN}$.

5.4.6. SSR vs. RDGCN

In alignment with HGCN, the performance of RDGCN was evaluated across three distinct versions of the dataset. Notably, on the $DBP15k_{FR\_EN}$, RDGCN exhibited superior performance relative to the other datasets and achieved higher values of the Hits@1 metric when compared with HGCN. This discernible advantage may be attributed to the incorporation of a dual-relation knowledge graph by RDGCN, which effectively harnesses the relational intelligence embedded within the knowledge graph.

By employing the RDGCN embedding module in conjunction with our proposed methodology, our approach has yielded an increase of approximately 1% in the $DBP15k_{ZH\_EN}$ and $DBP15k_{JA\_EN}$ datasets. However, in the $DBP15k_{FR\_EN}$ dataset, the improvement is only 0.27%.

5.4.7. Case Study

Figure 6 showcase the results of the experiments of both the baseline models and their respective SSR variants on the $DBP15k_{ZH\_EN}$ dataset. Each figure presents a horizontal axis denoting different types of node degrees and a vertical axis representing the number of entities. The blue squares depict the number of nodes that exhibit an increase in ranking for the actual cross-lingual equivalent entity within the candidate set upon rearrangement, denoted as "hit". On the other hand, the orange circles represent the total number of declines, referred to as "miss". Essentially, the blue squares indicate a positive impact of rearrangement, while the orange circles signify a negative effect. Consequently, the disparity between these two lines highlights the improvement effect of the SSR variant concerning the original model at different node degrees, where a larger difference denotes a more favorable improvement effect.

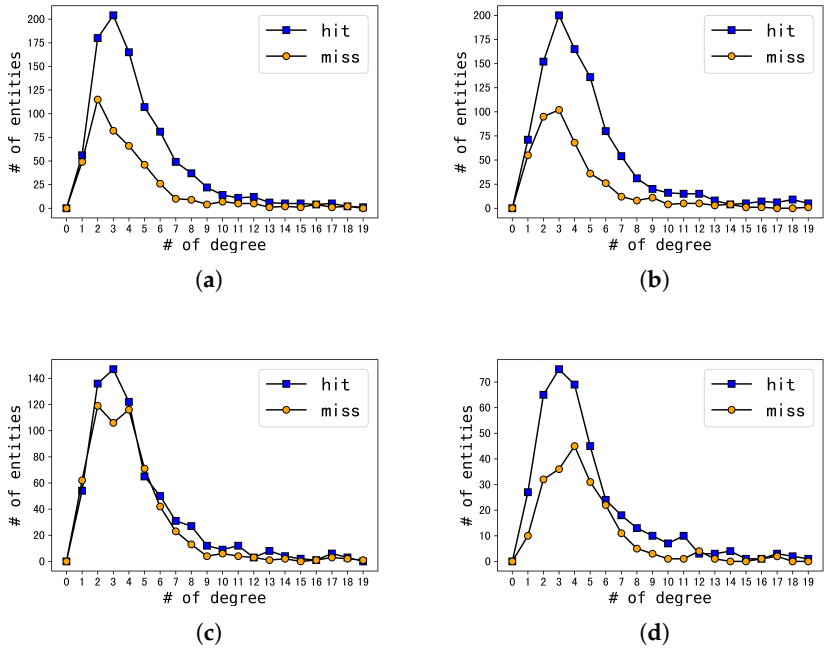


Figure 6. Influence via SSR. (**a**) SSR vs. GCNs; (**b**) SSR vs. JAPE; (**c**) SSR vs. HGCN; (**d**) SSR vs. RDGCN.

Upon careful examination of Figure 6, it becomes evident that the distribution of node degrees adheres to the statistical information of the dataset, as demonstrated in Figure 4.

Initially, we conducted a thorough analysis of graphs Figure 6. These graphs revealed that regardless of the degree of the node, ranging from 0 to 19, the total number of genuine cross-lingual equivalent entities in the candidate set that are ranked up after rearrangement consistently exceeded the total number of entities ranked down. Additionally, the difference between the number of hits and misses was always maintained at a higher value, with a minimum difference of 40. This phenomenon is the key factor responsible for the significant enhancement in the performance our method in comparison to the JAPE and GCN-Align embedding models.

Upon examining Figure 6, we observed that the disparity between the number of "hits" and "misses" was not substantial (the highest difference did not surpass 40) even in regions with the most densely distributed nodes. Notably, for the SSR variant of HGCN, there was only a significant difference in nodes with degrees of 2 and 3. Furthermore, as the node degree increased, the "miss" lines tended to exceed the "hit" lines.

To sum up, our proposed method has effectively enhanced the prediction outcomes for nodes with low degrees. However, the performance of the variant model shows a gradual deterioration for nodes with higher degrees.

5.4.8. Ablation Experiments and Sensitivity Analysis

In this paper, we propose a novel approach for computing the local structural similarity of nodes in a graph. Our method takes into account both the neighboring node information and the neighboring edge information present in the node's surrounding environment, which are denoted by the node information weight λ and the edge information weight δ in Equation (6), respectively. To demonstrate the significance of these two factors, we conduct ablation experiments and a sensitivity analysis in this section.

We present two sets of experiments to investigate the significance of neighboring node and edge information in our proposed approach. Specifically, we conducted the first set of experiments by maintaining $\delta = 0$, while varying the value of node weights λ to determine the effect of neighboring node information. The second set of experiments involved fixing

the node weights at $\lambda = 0$ and modifying the value of edge weights δ to examine the impact of neighboring edge information.

First, we investigate the significance of neighboring node information in the local structure of candidate set nodes. To this end, we perform experiments on several embedding modules, including JAPE, GCN-Align, AliNet, HGCN-JE, and RDGCN, by fixing the edge weight coefficient at $\delta = 0$ and adjusting the value of λ in the SSR variants. Each model is trained for 200 iterations, and the Hits@1 results are recorded and presented in Tables 7–11.

Table 7. Node weight for JAPE–SSR–5.

λ	$DBP15k_{ZH\_EN}$ *	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	39.39	33.84	28.75
0.5	43.72	36.10	**32.44**
1.0	**44.24**	**36.22**	32.11
1.5	43.54	35.54	31.47
2.0	43.42	35.23	31.34
2.5	42.90	34.77	30.95
3.0	42.90	34.77	30.95
3.5	42.59	34.44	30.84
4.0	42.59	34.44	30.84
4.5	42.27	34.30	30.68
5.0	42.27	34.30	30.38

* *Hits*@1 on DBP15k$_{ZH_EN}$.

The values of λ in the SSR variant of JAPE have been fine-tuned and the experimental outcomes are presented in Table 7. In the course of progressively augmenting the node weight λ from 0 to 5.0, the corresponding values of Hits@1 of the variant model exhibit a predominantly declining trend across all datasets. Optimal performance, as measured by Hits@1, is achieved when λ is set to 1 on the $DBP15k_{ZH\_EN}$ and $DBP15k_{JA\_EN}$ datasets. This finding indicates that in the local structure of nodes, the information emanating from neighboring nodes plays an equally crucial role as the embedding distance.

Table 8. Node weight for GCNs–SSR–5.

λ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	39.16	40.40	40.11
0.5	**43.63**	43.63	**42.41**
1.0	43.56	**44.15**	42.26
1.5	42.76	43.44	41.41
2.0	42.53	43.23	41.33
2.5	42.15	42.71	40.74
3.0	42.15	42.71	40.74
3.5	41.83	42.40	40.56
4.0	41.83	42.40	40.56
4.5	41.63	42.20	40.19
5.0	41.63	42.20	40.19

The experimental results of applying SSR on top of the GCN-Align embedding module are shown in Table 8. Similar to JAPE, the values of Hits@1 of the variant model on each dataset generally exhibit a decreasing trend as the node information weight λ is increased from 0 to 5.0, with the highest value achieved at $\lambda = 1.0$ only on the $DBP15k_{JA\_EN}$ dataset. These findings suggest that neighboring node information, in addition to the distance between node embeddings, plays a crucial role in the local structure of nodes when rearranging the nodes in the candidate set obtained from the original GCN-Align model, and such information significantly affects the performance of the final model.

Table 9. Node weight for AlN–SSR–5.

λ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	51.71	51.70	52.35
0.5	**53.86**	**53.19**	**54.57**
1.0	53.75	52.96	53.81
1.5	53.05	52.23	52.47
2.0	52.85	51.88	52.20
2.5	52.27	51.36	51.79
3.0	52.27	51.36	51.79
3.5	51.70	51.04	51.30
4.0	51.70	51.04	51.30
4.5	51.46	50.72	51.05
5.0	51.46	50.72	51.05

The experimental outcomes of applying SSR on top of AliNet are presented in Table 9. As the node information weight λ is gradually increased from 0 to 5.0, the variant model exhibits a consistent decreasing trend in the Hits@1 performance metric across all datasets. The best performance is achieved at $\lambda = 1.0$ for all datasets. This finding underscores the importance of incorporating neighboring node information in the local structure of nodes for the rearrangement of nodes in the candidate set generated by the original AliNet model.

The three sets of control experiments described above involved a uniform selection of λ values ranging from 0 to 5.0. Node embedding models such as JAPE and GCN-Align generate subpar embeddings for cross-lingual entity alignment and exhibit poor performance in this task. However, with the addition of SSR, the variant models are able to leverage both vector distance and node local structure similarity when searching for cross-lingual counterpart entities. Notably, the information of neighboring nodes in the node local structure plays an equally important role as the embedding vector distance. In contrast, AliNet, a structure-embedding model, considers the information of neighbors at a greater distance from the nodes. As such, the information of neighboring nodes in the local structure surrounding nodes can substantially improve the performance of the variant model when combined with SSR.

Table 10. Node weight for HJ–SSR–5.

λ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	69.31	76.09	88.95
0.05	69.68	76.52	89.35
0.10	70.30	77.03	**89.67**
0.15	70.76	77.74	89.39
0.20	70.82	77.84	89.35
0.25	**71.02**	**78.18**	88.79
0.30	70.52	77.69	88.17
0.35	70.08	77.35	87.54
0.40	70.05	77.30	87.44
0.45	69.89	77.10	86.97
0.50	69.89	77.10	86.97

The experimental results of combining SSR with the embedding module of HGCN-JE are presented in Table 10. As the node information weight coefficient λ is progressively increased from 0 to 0.50, the performance of the variant model exhibits an initial increase followed by a decrease, and the best performance is obtained when λ is set to 0.25, 0.25, and 0.10 on each dataset. These findings suggest that, although the original HGCN-JE model performs well, incorporating neighboring node information in the local structure of the nodes can further enhance the predictive accuracy of the variant model.

Table 11. Node weight for RG–SSR–5.

λ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	70.77	77.33	88.81
0.05	71.04	77.47	88.90
0.10	71.76	78.09	**89.06**
0.15	72.08	78.26	88.63
0.20	72.27	78.41	88.29
0.25	**72.53**	**78.54**	87.50
0.30	71.97	77.65	86.31
0.35	71.70	77.17	85.35
0.40	71.62	77.19	85.25
0.45	71.71	77.10	84.67
0.50	71.71	77.10	84.67

When applying our method to the original RDGCN model, as shown in Table 11, the node information weight coefficient λ is gradually increased from 0 to 0.50, resulting in a similar trend to HJ–SSR–5, where the variant model performance first increases and then decreases. The best results are achieved at $\lambda = 0.25$, $\lambda = 0.25$, and $\lambda = 0.10$ on each language version dataset, respectively. These experimental results demonstrate the critical role that neighboring node information in the local structure plays in positively influencing the prediction results and improving the overall performance of the model.

In comparison to the three control experiments mentioned earlier, the traversal interval of λ in our study is narrower. This is attributed to the competence of the two models which excel in learning high-quality knowledge graph embeddings. The prediction accuracy of cross-lingual corresponding entities, solely reliant on distance similarity, is already remarkably high. Consequently, the impact of neighboring nodes' information weights diminishes in comparison to the first three node-level models. Nonetheless, our proposed SSR variant manages to achieve discernible improvements over the original model.

The aforementioned experiments lead to the conclusion that the node information plays a crucial role in the model, significantly impacting its performance sensitivity to variations in λ.

After verifying the significance of the local structure node information, we proceeded to investigate the role of edge information. In this set of experiments, we set the node weight coefficient λ to 0 and uniformly sampled edge weight coefficients in the range $[0, 1.0]$. The experimental results of the variant model are presented in Tables 12–16. Since the Hits@1 results of the model on each dataset remain almost constant when varying the value of δ, we combine the results of different δ values.

The empirical findings indicate that the incorporation of neighboring edge information within the local structure makes a negligible contribution to the models' performance, with this parameter δ demonstrating low sensitivity.

Table 12. Edge weight for GCNs–SSR–5.

δ	$DBP15k_{ZH\_EN}$ *	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	39.16	40.40	40.11
$[0.1, 1.0]$	39.16	40.40	40.11

* *Hits*@1 on $DBP15k_{ZH\_EN}$.

Table 13. Edge weight for AlN–SSR–5.

δ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	51.71	51.70	52.35
$[0.1, 1.0]$	51.57	51.70	52.35

Table 14. Edge weight for JAPE–SSR–5.

δ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	39.39	33.84	28.75
[0.1, 1.0]	39.39	33.84	28.75

Table 15. Edge weight for RDGCN–SSR–5.

δ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	72.02	77.84	88.79
[0.1, 1.0]	72.02	77.84	88.79

Table 16. Edge weight for HJ–SSR–5.

δ	$DBP15k_{ZH\_EN}$	$DBP15k_{JA\_EN}$	$DBP15k_{FR\_EN}$
0	69.31	76.09	88.95
[0.1, 1.0]	69.31	76.09	88.95

6. Conclusions

In this paper, we propose a novel local structure-based rearrangement method that exploits the structural similarity between the source and target knowledge graphs and considers not only the distance between entity embeddings in the alignment phase, but also incorporates the feature of the local structure constituted by the entities and their surrounding nodes into the evaluation metric of entity similarity in order to improve the alignment accuracy, which is reflected in the rearrangement of the candidate set nodes. Experiments on three cross-lingual datasets of DBP15k demonstrate that the proposed approach can achieve promising performance. Through a comprehensive case study, ablation experiments, and a sensitivity analysis, we reveal that the information derived from neighboring nodes within the entity's local structure exerts a more substantial influence on the model and exhibits heightened sensitivity. Notably, our method demonstrates superior capabilities in handling nodes with smaller degrees. However, for nodes characterized by larger degrees and intricate structures, the improvement effect after rearrangement becomes less pronounced. This aspect poses a pivotal challenge, impeding the further enhancement in our method's performance.

Author Contributions: Conceptualization, G.L. and C.J.; methodology, G.L. and C.J.; software, G.L.; validation, G.L.; formal analysis, G.L., C.J. and L.S.; investigation, C.Y., J.Y. and J.S.; resources, C.Y.; data curation, G.L.; writing—original draft preparation, G.L.; writing–review and editing, G.L., C.J. and L.S.; visualization, G.L.; supervision, C.J. and C.Y.; project administration, C.J.; funding acquisition, C.J., J.S. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Zhejiang Science and Technology Plan Project (No. 2021C02060), the Natural Science Foundation of Zhejiang Province of China under Grant (No. LY21F020003), and the Scientific Research Foundation of Zhejiang University City College (No. X-202206). This research was also supported by the advanced computing resources provided by the Supercomputing Center of Hangzhou City University.

Data Availability Statement: The data in experiments is available on the Internet and can be accessed easily.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amoto, C.; Melo, G.D.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
2. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Neural Information Processing Systems (NIPS);* Neural Information Processing Systems Foundation: Lake Tahoe, NV, USA, 2013; pp. 2787–2795.

3. Ji, G.; Liu, K.; He, S.; Zhao, J. Knowledge graph completion with adaptive sparse transfer matrix. In Proceedings of the the 53th Conference of Association for Computational Linguistics (ACL), Phoenix, AZ, USA, 12–17 February 2016; Association for Computational Linguistics (ACL): Beijing, China, 2016; pp. 687–696.
4. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the the 28th AAAI Conference on Artificial Intelligence (AAAI), Québec City, QC, Canada, 27–31 July 2014; AI Access Foundation: Palo Alto, CA, USA, 2014; pp. 1112–1119.
5. Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; Liu, S. Modeling relation paths for representation learning of knowledge bases. *arXiv* **2015**, arXiv:1506.00309.
6. Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Zhao, D. Jointly learning entity and relation representations for entity alignment. *arXiv* **2019**, arXiv:1909.09317.
7. Pei, S.; Yu, L.; Yu, G.; Zhang, X. Rea: Rebust cross-lingual entity alignment between knowledge graphs. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2175–2784.
8. Knublauch, H.; Oberle, D.; Tetlow, P.; Wallace, E.; Pan, J.Z.; Uschold, M. A semantic web primer for object-oriented software developers. In *W3C Working Group Note*; WC3: Cambridge, MA, USA, 2006.
9. Suchanek, F.M.; Abiteboul, S.; Senellart, P. PARIS: Probabilistic alignment of relations, instances, and schema. *PVLDB* **2012**, *5*, 157–168. [CrossRef]
10. Shao, C.; Hu, L.M.; Li, J.Z.; Wang, Z.C.; Chung, T.; Xia, J.B. Rimom-im: A novel iterative framework for instance matching. *J. Comput. Sci. Technol.* **2016**, *31*, 185–197. [CrossRef]
11. L-Roby, A.E.; Aboulnaga, A. *ALEX: Automatic Link Exploration in Linked Data*; SIGMOD: Melbourne, Australia, 2015; pp. 1839–1853.
12. Hu, W.; Chen, J.; Qu, Y. *A Self-Training Approach for Resolving Object Coreference on the Semantic Web*; WWW: Hyderabad, India, 2011; pp. 87–96.
13. Zhuang, Y.; Li, G.; Zhong, Z.; Feng, J. *Hike: A Hybrid Human-Machine Method for Entity Alignment in Large-Scale Knowledge Bases*; CIKM: Singapore, 2017; pp. 1917–1926.
14. Chen, M.; Tian, Y.; Yang, M.; Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 1511–1517.
15. Sun, Z.; Hu, W.; Li, C. Cross-lingual entity alignment via joint attribute-preserving embedding. In Proceedings of the International Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Springer: Cham, Switzerland, 2017; pp. 628–644.
16. Sun, Z.; Hu, W.; Zhang, Q.; Qu, Y. Bootstrapping entity alignment with knowledge graph embedding. In Proceedings of the the 27th International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 4396–4402.
17. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2017**, arXiv:1609.02907.
18. Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Yan, R.; Zhao, D. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv* **2019**, arXiv:1908.08210.
19. Mao, X.; Wang, W.; Wu, Y.; Lan, M. From alignment to assignment: Frustratingly simple unsupervised entity alignment. *arXiv* **2021**, arXiv:2109.02363.
20. Jin, C.; Ruan, T.; Wu, D.; Xu, L.; Dong, T.; Chen, T.; Wang, S.; Du, Y.; Wu, M. HetGAT: A heterogeneous graph attention network for freeway traffic speed prediction. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–12. [CrossRef]
21. Gao, H.; Huang, J.; Tao, Y.; Hussain, W.; Huang, Y. The joint method of triple attention and novel loss function for entity relation extraction in small data-driven computational social systems. *IEEE Trans. Comput. Soc. Syst. TCSS* **2022**, *9*, 1725–1735. [CrossRef]
22. Gao, H.; Dai, B.; Miao, H.; Yang, X.; Barroso, R.J.D.; Hussain, W. A novel GAPG approach to automatic property generation for formal verification: The GAN perspective. *Acm Trans. Multimed. Comput. Commun. Appl. TOMM* **2023**, *19*, 16. [CrossRef]
23. Lisa, E.; Wolfram, W. Towards a definition of knowledge graphs. In Proceedings of the SEMANTICS 2016: Posters and Demos Track, Leipzig, Germany, 12–14 September 2016.
24. Dai, Y.; Wang, S.; Xiong, N.N.; Guo, W. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics* **2020**, *9*, 750. [CrossRef]
25. Maximilian, N.; Murphy, K.; Tresp, V.; Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **2015**, *104*, 11–33.
26. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-based multi-relational graph convolutional networks. *arXiv* **2020**, arXiv:1911.03082.
27. Hugh, G.; Jaffri, A.; Millard, I. Managing co-reference on the semantic web. In Proceedings of the WWW2009 Workshop: Linked Data on the Web (LDOW2009), Madrid, Spain, 1 January 2009.
28. Jiménez-Ruiz, E.; Grau, B.C.; Zhou, Y. LogMap: Logic-based and scalable ontology matching. In Proceedings of the International Semantic Web Conference, Bonn, Germany, 23–27 October 2011; Springer: Berlin, Heidelberg, 2011; pp. 273–288.
29. Lacoste-Julien, S.; Palla, K.; Davies, A.; Kasneci, G.; Graepel, T.; Ghahramani, Z. Sigma: Simple greedy matching for aligning large knowledge bases. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 572–580.
30. Hao, Y.; Zhang, Y.; He, S.; Liu, K.; Zhao, J. A joint embedding method for entity alignment of knowledge bases. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Beijing, China, 19–22 September 2016; pp. 3–14.

31. Wang, Z.; Lv, Q.; Lan, X.; Zhang, Y. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 349–357.
32. Wang, H.; Wei, Z.; Yuan, Y.; Du, X.; Wen, J.R. Exact single-source simrank computation on large graphs. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 11 June 2020; pp. 653–663.
33. Kriege, N.M.; Johansson, F.D.; Morris, C. A survey on graph kernels. *Appl. Netw. Sci.* **2020**, *5*, 6. [CrossRef]
34. Mao, X.; Wang, W.; Lan, M. Are negative samples necessary in entity alignment? An approach with high performance, scalability and robustness. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 26 October 2021; pp. 1263–1273.
35. Nguyen, T.T.; Huynh, T.T.; Yin, H.; Van, T.V.; Zheng, D.S.B.; Nguyen, Q.V. Entity alignment for knowledge graphs with multi-order convolutional networks. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 4201–4214.
36. Sun, Z.; Wang, C.; Hu, W.; Chen, M.; Dai, J.; Zhang, W.; Qu, Y. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 222–229. [CrossRef]