

Article

A Domain-Oriented Entity Alignment Approach Based on Filtering Multi-Type Graph Neural Networks

Yaoli Xu , Jinjun Zhong, Suzhi Zhang , Chenglin Li, Pu Li, Yanbu Guo , Yuhua Li, Hui Liang and Yazhou Zhang

School of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China

* Correspondence: yaolixu@zzuli.edu.cn (Y.X.); zhangsuzhi@zzuli.edu.cn (S.Z.)

Abstract: Owing to the heterogeneity and incomplete information present in various domain knowledge graphs, the alignment of distinct source entities that represent an identical real-world entity becomes imperative. Existing methods focus on cross-lingual knowledge graph alignment, and assume that the entities of knowledge graphs in the same language are unique. However, due to the ambiguity of language, heterogeneous knowledge graphs in the same language are often duplicated, and relationship triples are far less than those of cross-lingual knowledge graphs. Moreover, existing methods rarely exclude noisy entities in the process of alignment. These make it impossible for existing methods to deal effectively with the entity alignment of domain knowledge graphs. In order to address these issues, we propose a novel entity alignment approach based on domain-oriented embedded representation (DomainEA). Firstly, a filtering mechanism employs the language model to extract the semantic features of entities and to exclude noisy entities for each entity. Secondly, a Structural Aggregator (SA) incorporates multiple hidden layers to generate high-order neighborhood-aware embeddings of entities that have few relationship connections. An Attribute Aggregator (AA) introduces self-attention to dynamically calculate weights that represent the importance of the attribute values of the entities. Finally, the approach calculates a transformation matrix to map the embeddings of distinct domain knowledge graphs onto a unified space, and matches entities via the joint embeddings of the SA and AA. Compared to six state-of-the-art methods, our experimental results on multiple food datasets show the following: (i) Our approach achieves an average improvement of 6.9% on MRR. (ii) The size of the dataset has a subtle influence on our approach; there is a positive correlation between the expansion of the dataset size and an improvement in most of the metrics. (iii) We can achieve a significant improvement in the level of recall by employing a filtering mechanism that is limited to the top-100 nearest entities as the candidate pairs.

Keywords: domain knowledge graph; entity alignment; embedded representation; filtering mechanism



Citation: Xu, Y.; Zhong, J.; Zhang, S.; Li, C.; Li, P.; Guo, Y.; Li, Y.; Liang, H.; Zhang, Y. A Domain-Oriented Entity Alignment Approach Based on Filtering Multi-Type Graph Neural Networks. *Appl. Sci.* **2023**, *13*, 9237. <https://doi.org/10.3390/app13169237>

Academic Editors: Reza Shahbazian and Irina Trubitsyna

Received: 2 July 2023

Revised: 4 August 2023

Accepted: 6 August 2023

Published: 14 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A knowledge graph (KG) is a structured semantic network knowledge base that stores knowledge in the form of triples, and has powerful semantic expression capabilities in the fields of natural language processing, intelligent question answering, and intelligent recommendation [1]. Driven by the rapid development of the Internet, a wealth of domain knowledge graphs have been established, e.g., recipe knowledge graphs.

However, because most of the existing knowledge graphs have been established by independent institutions with different design concepts and requirements, there are differences in the described methods and descriptive focuses for the same object, resulting in the diversity and heterogeneity of data in the same domain knowledge graph [2]. However, different knowledge graphs also contain much complementary information. For instance, there are two recipe KGs, recipe 1 and recipe 2; both recipes include the dish named “braised pork”, which is described with regard to ingredients, cuisine, and cooking techniques. However, recipe 1 emphasizes the selection and preparation of ingredients, while recipe 2 emphasizes the cooking technique. When fusing the two recipes into a unified KG, it is

imperative to streamline and reorganize redundant information to optimize the structure of the KG. Additionally, the incorporation of complementary information can enhance the description of the “braised pork” and improve the overall quality and accuracy of the knowledge representation. Therefore, it is of great practical significance to study how to align the entities of heterogeneous knowledge graphs, in order to eliminate redundancy and integrate the knowledge graphs.

As an important research topic within natural language processing, entity alignment aims to discover the heterogeneous representations of the same real-world object among heterogeneous knowledge graphs, and to establish the equivalent-link relationships among them, so as to connect multi-source knowledge graphs to form a larger and richer knowledge base [3,4]. It is also known as entity resolution or entity matching [5,6]. In the process of integrating recipe KGs, we have to handle the following problems:

- Existing methods have to compute the embeddings of all the entities and relationships, but it is unnecessary to compute some entities which represent different real-world objects. Excluding these noisy entities can reduce the number of calculations and bring significant improvement to the accuracy of entity alignment.
- Existing works focus on embedding structural information; however, if there are few relationship connections between entities, this results in the sparse state of the entity neighborhood structure in domain KGs, especially for recipes.
- The traditional methods generally aggregate and propagate the attribute names of entities, but ignore attribute values. However, attribute value plays a significant role in enhancing entity alignment, and each attribute value has a different influence on an entity. It is non-trivial to utilize attribute values in order to enhance entity embedding.

However, existing methods cannot handle these problems, because these methods focus on enriching cross-lingual links [7–11], and assume that the descriptions of entities in knowledge graphs are consistent within the same language. In response to this, a novel entity alignment approach based on domain-oriented embedded representation (DomainEA) is proposed. DomainEA works as follows. In order to address the first problem, a filtering mechanism based on entity attributes first excludes the noisy entities that are impossible to align, and then selects the candidate set. For the second and third problems, a Structural Aggregator (SA) composed of Graph Convolutional Networks (GCNs) [12] employs attribute values to generate high-order neighborhood-aware embeddings for the entity structures. We introduce Graph Attention Networks (GAT) [13] into an Attribute Aggregator (AA) to effectively assign weights for attribute values according to their importance. Finally, a transformation matrix maps the embeddings of distinct KGs onto a unified space, and entity pairs that represent a real-world object are linked by calculating the similarity of their embeddings.

The main contributions of this paper are as follows:

- Considering that few relationship connections exist in domain-oriented embedded representation and the underuse of attribute values information, we employed a SA to generate high-order neighborhood-aware embeddings of entities through attribute values. Moreover, an AA utilizes the self-attention mechanism to dynamically calculate the weights between entities and attribute values and generates the attribute-aware embedding of entities. In addition, Multi-type Graph Neural Networks can enhance the aggregation of entity features.
- In order to exclude unnecessary computations and improve the accuracy of the entity alignment, we designed the filtering mechanism with entity attributes (e.g., taste or cooking technique) on domain KGs, which select the candidate set by the blocker. The experiments show that this can achieve expectations.
- Our approach only needs a few pre-aligned entities, and does not require any pre-aligned relationships or attributes between the KGs, which reduces the cost of manually annotating data in the early stages.
- The experimental results on a real-world dataset show that compared to the six state-of-the-art methods, our approach has higher accuracy and better stability.

2. Related Work

The existing entity alignment methods can be divided into three categories, namely statistics methods, classifier methods, and representation learning methods.

The core idea of the statistical methods is to find equivalent entities by comparing the similarity of entity names, entity attributes, and attribute values. In order to resolve the problems of a large amount of calculations and the method being limited to the calculation of the concept in ontology mapping, an improved mapping method of comprehensive ontology similarity [14] was proposed. This divides the large size of the original ontology into some smaller ontologies, and then computes the ontology conceptual similarity by combining three weighted similarities, which are derived from ontology concepts based on the How Net, semantics, and structure. Similar methods include PARIS [15] and RiMOM-IM [16]. Some methods [17] first calculate the similarity of each attribute via a variety of similarity measures, and then select the attribute similarity with the best performance to construct an integrated statistical model. The literature [18] uses the clustering model to train an adaptive distance function through the sample dataset, and through this function, the entities that may be aligned are clustered together. However, in heterogeneous knowledge bases it is difficult for the accuracy of statistical methods to reach a sufficient value, and such methods need to manually define some hyperparameters that quantify the degree of similarity between entities. It is clear that this kind of method requires the involvement of domain experts, incurring additional labor costs.

The core idea of the classifier methods [19,20] is to model the similarity of each attribute of the entity as a feature value, and then to transform the alignment problem into a two-classification or three-classification problem, and finally to complete the alignment task through the decision results obtained by the classifier model. These kinds of methods are decision trees, Support Vector Machines (SVM), ensemble learning algorithms, and so on. Some researchers [19] have proposed a decision model based on cost optimization, which produces a cost-optimal decision rule with the overall cost formula and Bayesian formula, and implements entity alignment according to the rule. The literature [20] employs machine learning techniques to analyze the semantic features of the associated dataset, and proposes an approach for aligning entities founded on the semantic features of the textual attributes related to the associated dataset. Compared to statistical methods, the classifier methods achieve a significant improvement in accuracy. However, such methods need to deliberately design different feature engineering strategies for different knowledge bases, so that the applicability of these classifier methods is limited and the portability is poor.

The representation learning methods mainly utilize the Knowledge Graph Embedding (KGE) technology, which learns the low-dimensional vector representations of the entities that capture their implicit features, and aligns the different entities which represent the same real-world object via these semantic embeddings. Such methods can automatically obtain entity pairs from heterogeneous knowledge graphs without a large number of artificial features, so that they can achieve a wider range of applications [5]. Representation-based learning methods can be subdivided into two categories. One is a semantic matching model, which assumes that a relationship between two entities can be translated as a vector in the bedding space. Based on the above hypothesis, TransE [21] is a translation mode which represents entity relationships in the knowledge base as translation operations of head and tail entities on low-dimensional vectors. Since then, various improved models, such as TransH [22], TransG [23], PTransE [24], and PTransR [25] have been proposed, which embed entities and relationships into different vector spaces, and can express more complex semantic relationships. Although this kind of method performs well in cross-language translation alignment problems, for domain knowledge graphs, the available structural information is frequently limited, which ultimately leads to sparsity in the vector representation, and which poses a challenge for the methods that heavily rely on this information.

The other is the Graph Neural Network (GNN), which learns the representation of the target entity by recursively aggregating the representation of adjacent entities. Graph

embedding methods mostly focus on the structural information and attribute information of the knowledge graphs. MuGNN [26] employs GNN to embed the structural information of the knowledge graph into multiple channels, then assigns weights to the relationships between the entities through the attention mechanism, and finally calculates the similarity of the embedding features to achieve alignment. HGCN [27] is an entity alignment method for jointly learning entity and relationship representations, and iteratively learns the embedding representations of entities and relationships. In addition, other methods, such as R-GCN [28] and FuAlign [29], embed the structural information of the entity and infer alignment relationships. In addition to using the structural information of entities for embedding, many scholars use the attribute information of entities to construct the feature vectors of entities. The literature [30] transforms the alignment task into a two-classification problem, then connects the attribute values of the entities to form a sequence of attribute values and, finally, uses the BERT classifier to make the final label prediction. However, it is difficult to completely learn the characteristics of entities by only using the structural information or attribute information of the knowledge base, so some methods which comprehensively utilize both structural and attribute information have been proposed. GCN-Align [31] utilizes GCNs to capture the structural and attribute information of entities and to generate high-quality embedding vectors. This method calculates the similarity of the structural feature vectors and attribute feature vectors of entities, and integrates these vectors through a weighted summation, which ultimately serves as the criterion for entity similarity assessment. AttrGNN [32] utilizes GNN to co-encode the structural and attribute information of entities, and assigns weights to different attributes of the entity using an attention mechanism. The representation learning methods have stronger robustness and better generalization performance.

In summary, compared to statistical and classifier methods, the entity alignment methods based on representation learning have the characteristics of needing less manual annotation, higher execution efficiency, and better knowledge expression ability. For the entity alignment task, learning the structural and attribute information of the entity at the same time is conducive to enhancing the feature representation of the embedded entity and improving the effect of entity alignment. However, in the process of attribute embedding, the traditional representation learning method only utilizes a single attribute or assigns all attributes with the same weight, and ignores important attribute information, which is non-trivial for entity alignment problems and is sensitive to irrelevant attribute information.

3. Problem Formalization

Let a knowledge graph be represented as $KG = (E, V, A, T)$, where E represents the collection of entities, V represents the collection of attribute values, A represents the collection of attributes, and T represents the collection of attribute triples. Each triple, $t = (h, a, v)$ in T , is represented as $T = \{(h, a, v) | h \in E, a \in A, v \in V\}$. Assume that two heterogeneous knowledge graphs are represented as $KG_1 = (E_1, V_1, A_1, T_1)$ and $KG_2 = (E_2, V_2, A_2, T_2)$. The goal of the entity alignment task is to infer the equivalent relationship $S = \{(e_{i1}, e_{i2}) | e_{i1} \in E_1, e_{i2} \in E_2, e_{i1} \equiv e_{i2}\}$, where \equiv indicates equivalence and i represents the i -th aligned entity pair. For example, in heterogeneous knowledge graphs KG_1 and KG_2 , the entity alignment problem is to infer whether the equivalent relationship between e_{i1} and e_{i2} exists or not, as shown in Figure 1. In the recipe field, the relationships between dishes are missing; therefore, we indirectly establish the associations of the dishes through the attribute values of the dishes. The green nodes represent entities and the white nodes represent attribute values.

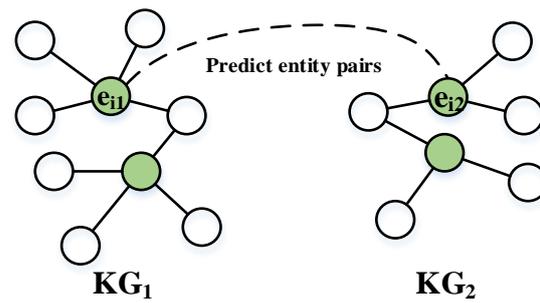


Figure 1. An entity alignment example.

4. The Filtering Multi-Type Graph Neural Networks

Throughout the alignment process for the recipes, we encountered issues because the different dishes have common ingredients, but the importance of identical ingredients varies among the different dishes, and there is no correlation between the dishes. The existing methods are not designed to address the aforementioned issues, and the utilization of these methods directly may not accurately and effectively identify all the entities of the same object. Given this, this approach proposes a novel domain-oriented entity alignment approach based on Filtering Multi-type Graph Neural Networks (DomainEA), which can effectively solve the above problems. DomainEA employs a filtering mechanism to filter confusing and unrelated entities to reduce unnecessary computations, and learns a more sophisticated embedded representation of both the structural level and attributive level in the recipe data through Multi-type Graph Neural Networks. As shown in Figure 2, the approach consists of a filtering module, embedding module, and alignment module.

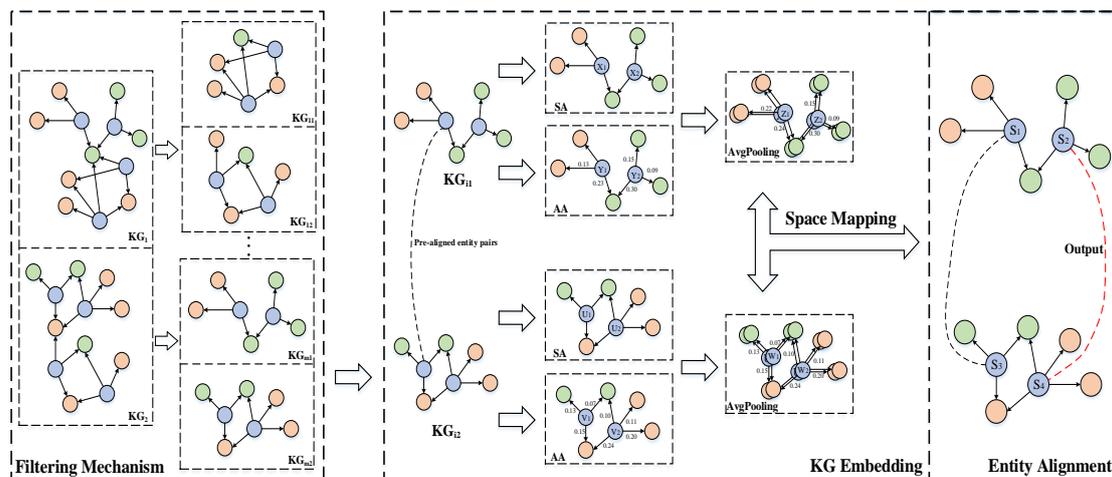


Figure 2. Filtering Multi-type Graph Neural Networks. We use a straight line with an arrow to represent the relationship between entities (blue nodes) and attribute values (orange or green nodes), with the arrow pointing from the entity to the attribute values. In KG_{i1} and KG_{i2} , i represents the candidate set for the i -th entity.

It works as follows. Firstly, considering that there are confusing entities and obviously misaligned entities, the filtering module employs pre-trained embeddings that consist of attributes and are trained on Wikipedia, in order to exclude noisy entities and to generate the candidate set of entities. The number of entities that are the input of embedding decreased dramatically. Secondly, in a certain domain, relationships between entities do not exist. For example, in the recipe, there is a lack of semantic relationships among the different dishes. The embedding module utilizes a SA to learn the structural-level embeddings via the attribute values, which can model the indirect relationships between entities. In addition, the same attribute value may play an important role in one entity,

but play a trivial role in other entities. For example, garlic plays a crucial role in certain dishes where it is the main ingredient, while in the majority of dishes garlic merely serves as a seasoning. We employ an AA to calculate the importance of attribute values in the form of weights, and to quantify these weights into the entity embeddings. According to the pooled embedding vectors of the pre-aligned entities, a space-mapping mechanism is designed to calculate a transformation matrix, which can transform two embedding vectors of different dimensions into a consistent vector. Finally, an alignment module is proposed to calculate the similarity of the embedding vectors using the Euclidean norm for entity alignment, and to align the different entities which represent the same real-world object.

4.1. The Filtering Module

The filtering module first employs a blocking mechanism to filter mismatched entities which are not deduplicated, and then utilizes a sub-knowledge graph generator to generate a set of sub-knowledge graphs. Figure 3 illustrates the filtering module in DomainEA. The blocking mechanism has shown a remarkable, outstanding performance in relational datasets [33] before the use of Multi-type Graph Neural Networks. The blocking mechanism designs a blocker to exclude the noisy entities, and to generate a candidate set for each entity in the original KGs. The sub-knowledge graph generator reconstructs the candidate sets of the entities into sub-knowledge graphs, which are the input of the embedding module.

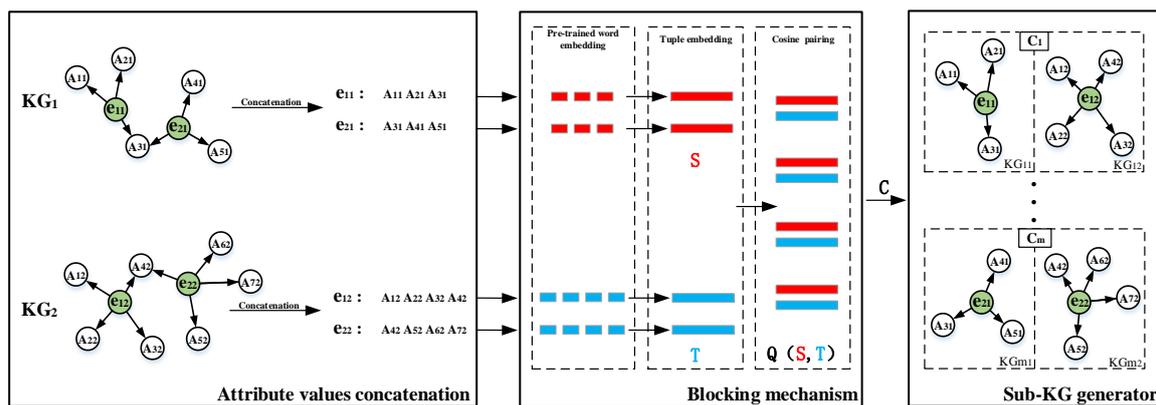


Figure 3. An example of the filtering module. C_m represents the candidate set for the m -th entity. KG_{m1} represents the sub-KG reconstructed by entities from the original KG_1 in the m -th candidate set.

Recently, as representation learning has become popular, many works have applied it to entity alignment [34–36] in KGs. In contrast, the blocking step has received far less attention; only a few recent works have applied it to generate candidate sets [37–40]. If we embed two heterogeneous knowledge graphs directly without a blocker in our approach, there are a large amount of entities that would need to be matched and would result in introducing noise. Moreover, directly computing all the entity embeddings of two heterogeneous knowledge graphs could take a lot of time in the subsequent calculation process; a considerable amount of the time spent would be unnecessary, which could reduce the efficiency of the entire model. Although the composition of the same entity in heterogeneous KGs is not identical, the same attributes may still exist for the same entity. In our work, we divide the attributes of an entity into single-value attributes and multi-values attributes. In addition to multi-values, such as the ingredients, each dish also has some single-value attributes, such as cooking techniques and taste. However, these single-value attributes are unsuitable for the embedding of a SA and an AA, but they can construct a well-performing blocking mechanism that excludes noisy entities and generates candidate sets for the embedding module. The blocking mechanism not only simplifies the calculation process, but also eliminates mismatched entities in advance, and enhances the final entity alignment.

For the recipe, we build the blocking mechanism with attribute-specific semantics of the entities in the KGs. These attribute-specific semantics are derived by analyzing the single-value attributes of the entities, such as cooking techniques, taste, and cooking difficulty. We first concatenate all the single-value attribute values of each entity in the KGs into a string. For example, in Figure 3, all the single-value attribute values of e_{11} consist of A_{11} , A_{21} , and A_{31} . We contact A_{11} , A_{21} , and A_{31} into a concatenated string, which is fed into the pre-trained word embedding. The blocking mechanism consists of the pre-trained word embedding, the tuple embedding, and the cosine pairing, as shown in Figure 3. The pre-trained word embedding employs fastText to generate a high-dimensional vector representation of each word in the concatenated string. Subsequently, the tuple embedding aggregates these vectors into two vector sets, denoted as S and T , in which each embedding has the same dimension and represents the concatenated string. The cosine-pairing module selects the top-K nearest neighbors by calculating the similarity $Q(S, T)$, and generates candidate sets, C , for the embedding module.

4.2. The Embedding Module

The embedding module mainly performs one hot encoding and learns the embedding of features for the entities. Our approach incorporates a SA and an AA into the embedding module. Firstly, the SA captures the structural information of the knowledge graph by employing three hidden layers, which is effective in addressing the limited connectivity between entities in the domain KGs. Secondly, the AA assigns different weights according to the importance of the attribute values through the attention mechanism, which can enrich the representation of entities. Finally, jointly embedding both structural and attributive information can capture more complex and diverse relationships among the entities and improve the accuracy of our approach. Table 1 outlines the parts of the parameters of our approach.

Table 1. The parts of the parameters of the SA and AA.

Notation	Description
σ	The nonlinear activation function
A	The adjacency matrix
I	The identity matrix
D	The degree matrix corresponding to the adjacency matrix
$W^{(l)}$	The weight coefficient matrix of the l -th layer
$\hat{A} = A + I$	The adjacency matrix with self-connection
N_i	The collection of neighboring entities of the i -th entity in KGs
H	The embedding of domain KGs
$b^{(l)}$	The bias term of the l -th layer

4.2.1. The Structure-Aware Entity Embedding

There is a lack of relationships between the entities in the domain KGs. To address this issue, we utilize the attribute values (represented by white dots) as the bridges and conduct iterative computations to progressively learn the multi-hop structural features of the entities through the multiple hidden layers of the SA. We first construct the input of the SA according to the attribute triples. Subsequently, we employ the SA with multiple hidden layers to learn the high-dimensional representation vectors for each entity and attribute value. These representation vectors can capture the semantic information between entities and attribute values, and the relevant information between the entities gradually spreads to the representation vectors of the other entities through the virtual relationship of attribute values. Finally, the indirect relationship (represented by a dotted green line with an arrow) between the entities can be captured through these representation vectors. The aggregation process of the SA is shown in Figure 4.

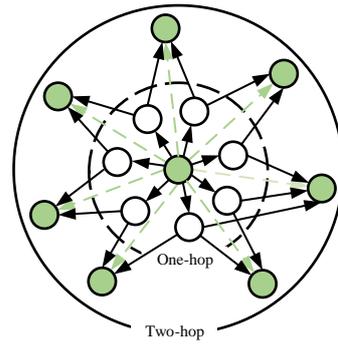


Figure 4. The diagram of the structural aggregator. The black line with an arrow represent the relationships between entities and attribute values, and the dotted green line with an arrow represent the indirect relationships between the entities.

The computation of the hidden layer plays a crucial role in the aggregation process of the SA. Due to the sparsity or even absence of relationship triplets in the domain knowledge graphs, we utilize attribute triplets as a substitute for relationship triplets for feature embedding. Multiple layers of the SA are employed to indirectly establish associations between entities by learning multi-hop neighborhood information. The neighborhood information can be quantified in the form of multiple hidden layers. Equation (1) encapsulates a comprehensive formulation for the update of the l -th hidden layer, $H^{(l)}$, within the SA. The function, $f(\cdot)$, is denoted as the update mechanism that utilizes the input from the previous layer, $H^{(l-1)}$, and the adjacency matrix, A .

$$H^{(l)} = f(H^{(l-1)}, A) \tag{1}$$

In Equation (1), the parameters of the feature matrix, $H^{(0)}$, and the adjacency matrix, A , in the SA are initialized according to the information about the dishes and related ingredients. Equation (2) presents a specific instantiation of the update mechanism, where an activation function, $\sigma(\cdot)$, is applied and the propagation of information from the previous layer, denoted as $AH^{(l-1)}W^{(l-1)}$, is incorporated. In addition, our approach selects LeakyReLU as the specific activation function to mitigate gradient vanishing.

$$H^{(l)} = \sigma(AH^{(l-1)}W^{(l-1)} + b^{(l-1)}) \tag{2}$$

In the equation above, $b^{(l-1)}$ represents the bias, which is a parameter adjusted through back-propagation. To effectively capture the bidirectional relationships among the entities and the inherent features of the entities during the process of feature learning, we perform symmetric normalization and self-connection processing on the adjacency matrix, A , as indicated by Equation (3):

$$\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} + I \tag{3}$$

Equation (4) is a variation of Equation (2). The update mechanism in Equation (4) captures more structural information of the KGs through incorporating the normalized Laplacian $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$, where \hat{D} is the diagonal matrix of node degrees. To accomplish this, the propagation of information from the preceding layer is rescaled by employing the diagonal matrix, $\hat{D}^{-\frac{1}{2}}$. This rescaling strategy ensures a balanced influence among neighboring entities in the knowledge graphs. By iteratively solving Equation (4), the model is able to learn the higher-order features of the entities.

$$H^{(l)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l-1)}W^{(l-1)} + b^{(l-1)}) \tag{4}$$

Through Equation (4), the embedding features of entity $h_i^{(l)}$ in the l -th layer can be derived as Equation (5), which illustrates the relationship between the embedding features of entity $h_i^{(l)}$ and its adjacent entities.

$$h_i^{(l)} = \sigma \left(\sum_{j \in N_i} \widehat{D}^{-\frac{1}{2}} \widehat{A} \widehat{D}^{-\frac{1}{2}} h_j^{(l-1)} \mathbf{W}^{(l-1)} + b^{(l-1)} \right) \quad (5)$$

4.2.2. The Attribute-Aware Entity Embedding

The existing methods only employ attribute names in attribute embedding, but overlook attribute values and the weights of the attribute values. Domain knowledge graphs, especially of recipes, encompass a large number of attribute values that describe entities with rich details. In addition, some attributes play a non-trivial role among the attributes of entities. For example, the dish “tomato scrambled eggs” is mainly composed of tomatoes and eggs, not salt. An AA is proposed to generate attribute-aware embeddings of the entities via their attribute values and attribute weights. The AA employs the attention mechanism and assigns appropriate weights to entity relationships based on the influence of adjacent entity features on the current entity. The AA requires paired adjacent entities during training, and we utilize a multi-headed AA to improve the generalization expression ability.

The computational process of the AA can be decomposed into two parts. The first part is to calculate the normalized attention coefficient of the entities. To obtain the normalized attention coefficient of the j -th neighboring entity, we first compute the attention coefficient according to Equation (6). Subsequently, we employ softmax to normalize the result, as shown in Equation (7).

In Equation (6), we first employ the weight matrix, \mathbf{W} , to perform a linear transformation on the features of entities h_i and h_j , and then concatenate the transformed result. This process can map the original features onto a higher-dimensional feature space, which facilitates the capture of intricate inter-entity relationships. Next, we utilize the attention weight vector, a^T , to assign the adaptive weight for different features via computing the scalar product with the concatenated result. Finally, we apply the LeakyReLU activation function to the computed result to obtain the final attention score, e_{ij} .

$$e_{ij} = \text{LeakyReLU} \left(a^T [\mathbf{W}h_i || \mathbf{W}h_j] \right) \quad (6)$$

$$a_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}, \quad j, k \in N_i \quad (7)$$

The second part aggregates the attention and characteristics of neighboring entities, and achieves the comprehensive attribute-aware embeddings of the entities. The comprehensive attribute-aware embedding of the i -th entity denoted as h'_i , is calculated as Equation (8).

$$h'_i = \sigma \left(\sum_{j \in N_i} a_{ij} \mathbf{W}h_j \right) \quad (8)$$

In order to comprehensively and diversely capture the representation of KGs, our approach utilizes a multi-headed AA to capture the features of the entities from various subspaces. In particular, each attention head independently learns its own attention weights and generates corresponding feature vectors. K represents the number of heads of the AA in Equation (9).

$$h'_i(K) = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in N_i} a_{ij}^k \mathbf{W}^k h_j \right) \quad (9)$$

4.2.3. The Jointly Embedding

In order to mitigate overfitting and enhance the accuracy and efficiency of DomainEA, it is imperative to reduce and aggregate the embedding spaces generated by the SA and AA in the jointly embedding phase. A global average pooling strategy is proposed to merge the embedded spaces of the SA and AA. This strategy can not only capture spatial information effectively, but also exhibits greater resilience to input space variations. Specifically, the global average pooling strategy consolidates spatial information by summing it up and is thus less sensitive to changes in the input space. Figure 5 illustrates the schematic diagram of the jointly embedding phase.

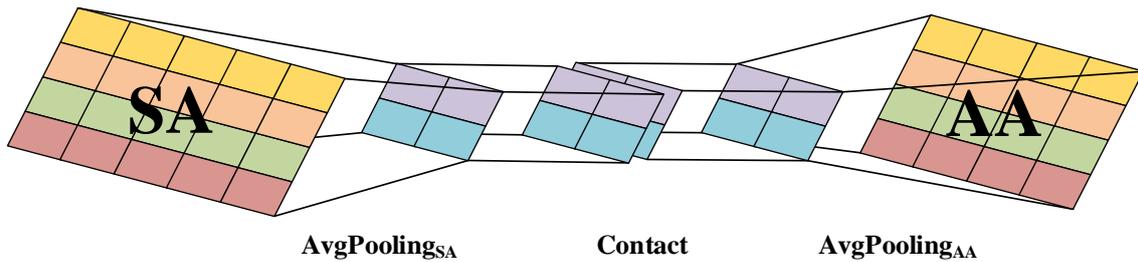


Figure 5. A schematic diagram of the jointly embedding phase.

The purpose of the jointly embedding phase is to contact the comprehensive embeddings of the KGs, which is computed according to Equation (10). In Equation (10), we employ the average pooling to extract the salient features from the output of aggregation of $H_{SA}^{(l)}$ and $H_{AA}^{(l)}$, and contact them to generate the jointly embedding. $H_{SA}^{(l)}$ and $H_{AA}^{(l)}$, respectively, represent the l -th output of the SA and AA embedding mechanisms.

$$H^{(l)} = Contact\left(AvgPooling\left(H_{SA}^{(l)}\right), AvgPooling\left(H_{AA}^{(l)}\right)\right) \quad (10)$$

4.2.4. The Spatial Mapping

After the jointly embedding phase, the entity embeddings of the distinct knowledge graphs may have different dimensions, since the entities from distinct knowledge graphs have different numbers of attribute values. The spatial mapping phase is tasked with mapping the embedding spaces of two distinct knowledge graphs onto a unified space. There are some pre-aligned entity pairs, which are selected before learning, and have embedding vectors after the above-mentioned embedding phases. Through the embedding vectors of the pre-aligned entity pairs from different KGs, we can achieve the transformation matrix, T , calculated by Equation (11). $h_{G_1}^n$ and $h_{G_2}^n$, respectively, represent the embedding eigenvectors of the n -th pair of matching entities between KG_1 and KG_2 . T represents the transformation matrix of the two embedding vector spaces.

$$\begin{cases} h_{G_1}^1 = Th_{G_2}^1 \\ h_{G_1}^2 = Th_{G_2}^2 \\ \cdot \\ \cdot \\ h_{G_1}^n = Th_{G_2}^n \end{cases} \quad (11)$$

4.3. The Alignment Module

After the embedding module, the entity embeddings of the heterogeneous graphs have a uniform vector space. The alignment module realizes the alignment task by calculating

the similarity metric between the embedding features. Our approach utilizes the L_2 norm to calculate the distance between pairs of candidate entities, which is depicted in Equation (12):

$$D(e_i, e_j) = \|h(e_i) - h(e_j)\|_2 \quad (12)$$

The variables e_i and e_j , respectively, represent pairs of candidate entities in two heterogeneous graphs, while $D(e_i, e_j)$ signifies the distance of the candidate entities between e_i and e_j . Respectively, $h(e_i)$ and $h(e_j)$ represent the eigenvectors of the embedded entities, e_i and e_j .

Our approach falls under the category of an unsupervised entity alignment method, so we introduce an unsupervised alignment consistency loss for the approach. Firstly, we extract the set of all neighboring entities, e_j , connected to e_i in KG_1 , as N_i . Similarly, we identify the corresponding entity e'_i in KG_2 , and collect its neighbouring entities e_k to form N'_i . Subsequently, by evaluating the difference term for each entity pair (e_i, e'_i) using Equation (13) and aggregating the results, we obtain the alignment discrepancy.

$$\mathcal{L} = \sum_{e_i \in KG_1} \left\| \frac{1}{|N_i|} \sum_{e_j \in N_i} D(e_i, e_j) - \frac{1}{|N'_i|} \sum_{e_k \in N'_i} D(e'_i, e_k) \right\| \quad (13)$$

According to Equation (13), it encourages similar entities to possess similar neighboring entities in both knowledge graphs while maintaining consistency in their alignments. Ultimately, by minimizing the consistency loss, our approach can learn a consistent alignment pattern and facilitate the alignment of entity pairs.

5. Experiments and Results

5.1. The Experimental Settings

5.1.1. Datasets

Most existing works evaluate their methods on DBP15K, which can only be used for the cross-lingual KG alignment task and is not suitable for our research. DomainEA is proposed for aligning entities of different sources to improve the quality of domain knowledge graphs, so we conducted experiments on two heterogeneous datasets, e.g., ZH-MSJ and ZH-ZHYSW, from Chinese recipe websites. Notably, the dataset has been released on GitHub <https://github.com/ZhongJinjun/DomainEA> (accessed on 31 December 2022) to facilitate access for scholars who may wish to employ it for research purposes. ZH-MSJ consists of 8156 entities, 18 attributes, and 95,145 attribute triples. ZH-ZHYSW includes 6083 entities, 11 attributes, and 46,652 attribute triples, as shown in Table 2. The original knowledge graphs are constructed according to ZH-MSJ and ZH-ZHYSW.

Table 2. Statistics of datasets.

Dataset	Entity	Attribute	Attribute Triple
ZH-MSJ	8156	18	95,145
ZH-ZHYSW	6083	11	46,652

5.1.2. Experimental Environment

This section describes the experimental environment for DomainEA, including the hardware and software tools, shown in Table 3, and the hyperparameters of the approach, shown in Table 4. We ultimately set the dimensional embeddings of the SA and AA at 128.

5.1.3. Evaluation Metrics

The state-of-the-art methods were evaluated in terms of Hits@n and Mean Reciprocal Rank (MRR). For a fair comparison, we employed Hits@n and MRR to evaluate all the experimental results. Hits@n indicates the proportion of the target entities which are ranked in the top-n positions, as shown in Equation (14). MRR indicates the average reciprocal

rank of the target entity in the alignment ranking, as shown in Equation (15). For all the evaluation metrics, a higher score indicates better performance. We report Hits@1, Hits@10, Hits@50, and MRR for the approach on each dataset.

$$\text{Hits}@n = \frac{1}{N} \sum_{i=1}^N I(\text{Rank}_i \leq n) \quad (14)$$

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{\text{Rank}_i} \quad (15)$$

Table 3. Hardware and software tools.

Tools	Value
CPU	Intel Core i7-12700H
RAM	32 GB
HDD	2 TB
PyTorch	1.10.2
Python	3.6.2
Scipy	1.5.4
Numpy	1.16.2

Table 4. Hyperparameters of DomainEA.

Hyperparameters	Description	Value
Keep-prob	The keep probability of neuron	0.9
Learning rate	The step size for model parameter updates	0.001
Alpha	The slope of LeakyReLU	0.2
Max epoch	The maximum number of training epochs	100

In the filtering module, we excluded noisy entities and generated candidate sets, C , on Tables 1 and 2. We evaluated three blocking solutions in terms of Recall and CSSR on ZH-MSJ and ZH-ZHYSW. A higher Recall and lower CSSR indicate a better performance. The Recall and CSSR, respectively, are shown in Equations (16) and (17). G represents the set of true matches between A and B .

$$\text{Recall} = \frac{|G \cap C|}{|C|} \quad (16)$$

$$\text{CSSR} = \frac{|C|}{|A \times B|} \quad (17)$$

5.1.4. Baseline Methods

We compared our approach (DomainEA) to the following baselines:

- GCN-Align: GCN-Align [31] employs GCN to effectively encode both the structural and attribute information of the entities and to generate high-quality embedding vectors. The method calculates the similarity of the structural and attribute feature vectors of the entities, and subsequently integrates them via a weighted summation, which provides a criterion for the entity similarity assessment.
- MuGNN: MuGNN [26] employs GNN to embed the structural information of the knowledge graph into multiple channels. Additionally, it utilizes an attention mechanism to assign weights to the relationships between entities, ultimately facilitating the calculation of the similarity of embedding features for achieving alignment.
- HGCN: HGCN [27] is an entity alignment method for employing GCN to capture the implicit features of entities and relationships via jointly learning, and to iteratively learn the embedding representations of entities and relationships.

- FGWEA: FGWEA [41] is an unsupervised entity alignment framework with Gromov–Wasserstein distance. The method can make full use of the structural information of the knowledge graph to realize a comprehensive comparison of the corresponding entities in different knowledge graphs through optimizing entity semantics and the knowledge graph structure.
- PEEA: PEEA [42] belongs to a weakly supervised learning framework. In addition to absorbing structural and relational information, PEEA is designed to enhance the connections between distant entities and labeled entities by integrating positional information into the representation learning process through a Position Attention Layer (PAL).
- SDEA: SDEA [43] consists of attribute embedding and relation embedding. SDEA first employs the pre-trained language model transformer to extract the semantic information of attribute values, and then utilizes GRU equipped with an attention mechanism to aggregate the structural information of neighbor entities.

5.2. The Experimental Results

5.2.1. The Comparison of Six Baseline Methods

We conducted comparative experiments on ZH-MSJ and ZH-ZHYSW. The experimental results of all the metrics are shown in Table 5. Most of the existing methods assume that the knowledge graph has abundant relationships among the entities. However, the absence of relationships between entities in the domain knowledge graph undermines the efficacy of the baselines and the accomplishment of the alignment task properly. Consequently, we utilized attribute triples to replace relationship triples in the learning process of the baselines, and denoted them with the symbol, “+”. “↑” and “↓” indicates an increment or decrement in the baselines, which utilized attribute triples to replace relationship triples compared to themselves in Table 5. We could not execute MuGNN and HGCN directly, for they mainly utilize relationship triples for embedding, so we utilized “—” to represent their values in the metrics. Moreover, GCN-Align employs GCN to embed structural and attribute information for the entities, so GCN-Align+ is similar to GCN-Align in all the metrics.

Table 5. Comparative experiments.

Method	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
GCN-Align	46.25	86.74	92.21	60.07
MuGNN	—	—	—	—
HGCN	—	—	—	—
FGWEA	55.94	88.79	92.36	69.45
PEEA	53.59	88.61	90.16	67.98
SDEA	55.18	89.26	91.65	70.94
GCN-Align+	47.03 (0.78↑)	86.13 (0.61↓)	92.73 (0.52↑)	60.41 (0.34↑)
MuGNN+	53.71	92.48	96.74	69.09
HGCN+	59.02	95.38	97.19	73.23
FGWEA+	62.14 (6.2↑)	97.24 (8.45↑)	99.02 (6.66↑)	76.61 (7.16↑)
PEEA+	60.39 (6.8↑)	93.87 (5.26↑)	98.71 (8.55↑)	74.63 (6.65↑)
SDEA+	62.98 (7.8↑)	97.45 (8.19↑)	98.15 (6.5↑)	78.77 (6.83↑)
DomainEA	64.66	98.07	98.91	79.02

Table 5 suggests that the performance of DomainEA outperforms the baselines of almost all the metrics. The reasoning for that is, in the filtering module, DomainEA can effectively exclude noisy entities and select a few candidate entities before KG embedding. In the embedding module, we employed the SA to generate the high-order local features, and introduce attention to the dynamically calculated weights between the entities and attribute values in the AA. These strategies enhanced the efficacy of entity alignment. However, in comparison to the other baselines, MuGNN+ and HGCN+ only rely on structural information for embedding, while ignoring the significance of entity attribute values on

the alignment tasks, which results in the loss of semantic information about the entities. Although GCN-Align+ jointly embeds both the structural and attribute information of the KGs, as domain-specific KGs contain abundant interference information, GCN-Align+ cannot effectively address this problem, which leads to a suboptimal alignment performance. PEEA+ relies on a limited number of anchor links for training, but it is a challenge to obtain high-quality anchored links in recipes, so the performance of PEEA+ may be limited. FGWEA+ mainly utilizes structural information to optimize entity semantics and knowledge graph structure. However, in domain knowledge graphs, the lack of structural information makes us have to utilize attribute triples to replace the relationship triples, which limits the effectiveness of FGWEA+. SDEA+ demonstrates sub-optimal performance across all the metrics. A thorough analysis of its limitations reveals that it employs a transformer to extract semantic information from the entity attributes, and utilizes a GRU with an attention mechanism to capture the structural information of entities. While this design yields satisfactory results for the entity alignment task of the domain knowledge graph, it falls short in effectively filtering out noisy entities. As a consequence, SDEA+ slightly underperforms compared to DomainEA.

Moreover, the results reveal a substantial enhancement in the performance of the baselines when we employed attribute triples to replace relationship triples for entity embedding in the entity alignment task of the domain knowledge graph. Specifically, FGWEA+, PEEA+, and SDEA+ achieved a 5.26–8.55% improvement in all metrics. In particular, FGWEA+, respectively, achieved improvements of 6.2%, 8.45%, 6.66%, and 7.16% in Hits@1, Hits@10, Hits@50, and MRR. PEEA+ achieved an average improvement of 6.8% across all metrics, with a particularly substantial improvement of 8.55% observed in Hits@50. SDEA+, respectively, achieved improvements of 7.8%, 8.19%, 6.5%, and 6.83% in Hits@1, Hits@10, Hits@50, and MRR. This improvement substantiates the indispensability of integrating attribute triples into the process of embedding.

5.2.2. Ablation Experiments

For a comprehensive evaluation of our approach, we conducted three comparative experiments. Respectively, the first experiment was to compare the three deep-learning blocking methods. These methods have different tuple embedding solutions in the filtering module, and these solutions, respectively, are autoencoder, CTT, and hybrid. The second experiment was to evaluate DomainEA and DomainEA-f; DomainEA-f refers to an entity alignment method which does not have a filtering module compared to DomainEA. The third experiment was to evaluate the effect of DomainEA with various hidden layers in the SA. The primary objective was to verify the necessity of implementing the filtering module and to investigate the influence of the number of hidden layers in the SA on the experimental outcomes.

In the filtering module, we employed self-supervision deep-learning blocking to exclude noisy entities and generate the candidate sets for each entity. Blocking consisted of pre-trained word embedding, tuple embedding, and cosine pairing. We selected three representative solutions in the tuple embedding: autoencoder, CTT, and hybrid, which achieve higher recall and smaller candidate sets on the various datasets. These tuple embedding solutions are shown as follows:

- Autoencoder: The autoencoder model consists of an aggregator, encoder, and decoder. The aggregator can dispose of the various sequences of word embedding, which the feed-forward NN cannot accept. The encoder and decoder utilize two-layer feed-forward NNs with the Tanh activation function to reconstruct the feature of the entity.
- CTT: The CTT model consists of an aggregator, Siamese summarizer, and classifier. CTT exploits information in multiple tuples. It perturbs the tuples in Tables 1 and 2, and generates synthetic labeled data, which can train a deep-learning model to produce embeddings for each tuple.

- **Hybrid:** The hybrid model consists of an autoencoder and CTT, which are stacked by training the autoencoder first and then the CTT. This method employs the trained encoder of the autoencoder as the aggregator for CTT to generate the tuple embeddings.

We conducted an evaluation of the recall of the three methods above on various candidate set sizes. The results, presented in Table 6, indicate that the autoencoder almost always achieved the highest recall on multiple candidate sets, compared to the CTT and hybrid. Moreover, the hybrid model, respectively, achieved the highest recall on 1.41 and 2.53, and both autoencoder and CTT achieved the highest recall on 5.62. In general, all three methods had high recall on small candidate set sizes. Therefore, our approach selected autoencoder as the tuple embedding for the filtering module.

Table 6. The recall of solutions on various CSSR.

CSSR (%)	Autoencoder	CTT	Hybrid
1.41	81.03	78.41	83.26
1.69	84.74	81.69	84.39
1.97	87.16	84.69	86.40
2.25	92.63	87.49	89.92
2.53	94.25	93.82	96.20
2.81	98.03	96.62	97.05
5.62	99.90	99.90	99.57

We conducted a comparative experiment on DomainEA-f and DomainEA. Through this experiment, we could evaluate the effectiveness of the filtering module.

Table 7 reveals that DomainEA outperformed DomainEA-f on all the evaluation metrics and has superior runtime efficiency. Specifically, DomainEA, respectively, achieved improvements of 6.27%, 2.34%, 1.25%, and 7.88% in Hits@1, Hits@10, Hits@50, and MRR, while reducing the runtime by 43.04 s when compared to DomainEA-f. The result shows that DomainEA can effectively exclude noisy entities and greatly simplify the calculation process.

Table 7. Entity alignment results with a change of filtering module.

Method	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)	Time (s)
DomainEA-f	58.39	95.73	97.66	71.14	49.47
DomainEA	64.66	98.07	98.91	79.02	6.43

We conducted an experiment with various numbers of hidden layers of the SA and investigated the impact of the number of hidden layers on our approach. The experiment results are shown in Table 8.

Table 8. Entity alignment results with various numbers of hidden layers of the SA.

#Layers	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
1	53.88	93.98	97.74	71.91
2	64.32	97.24	98.16	77.72
3	64.66	98.07	98.91	79.02
4	64.41	97.91	98.93	78.86
5	62.32	97.15	98.33	76.35

We can infer from the experimental results in Table 8 that increasing the number of hidden layers in the SA has a positive impact on all the evaluation metrics, but does not mean better performance will be achieved. In particular, the 1-layer SA received the lowest scores on all the metrics compared to the others. Except for the 4-layer SA, which recorded the highest score in Hits@50, the 3-layer SA achieved the highest scores on the

other performance metrics. However, for the metric Hits@n, for the same score, the smaller the value of n was the more superior the method. Moreover, when the number of layers of the SA exceeded three, the performance of DomainEA declined. Overall, these findings suggest that increasing the SA layer can capture the higher-order entity features, but the model will incur an overfitting problem as the number of SA layers increases. It is important to select the appropriate number of hidden layers of the SA for the specific application.

5.2.3. Results on Various-Scale Datasets

We conducted an experiment with various sizes of datasets to evaluate the stability of our approach. We randomly selected 1000, 2000, and 3000 entities from the original KGs to construct three sub-KGs, noted as KG_1K, KG_2K, and KG_3K, in order to evaluate DomainEA and observe the trend of the change in metrics. The experimental results are depicted in Table 9. We can observe the trends of variation for all the metrics. Firstly, Hits@1 exhibits fluctuations with the increasing size of the dataset. From KG_1K to KG_2K, Hits@1 decreases by 0.35 percentage points. However, Hits@1 and MRR achieve significant improvement on KG_3K, reaching 60.48% and 75.22%. Secondly, in relation to Hits@10 and Hits@50, there is a positive correlation between the expansion of the dataset size and the improvement in these metrics. For instance, Hits@10 rises from 95.37% to 95.98%, and Hits@50 escalates from 97.48% to 98.16%. In summary, as the size of the dataset changes, the fluctuations of all the evaluation metrics are small. It implies that the various sizes of datasets have a subtle influence on the evaluation metrics and our approach has favorable stability.

Table 9. Entity alignment results with various numbers of hidden layers of GCNs.

Dataset	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
KG_1K	57.24	95.37	97.48	74.35
KG_2K	56.89	95.48	97.66	73.81
KG_3K	60.48	95.98	98.16	75.22

6. Summary

In this work, we focus on the domain problem of entity alignment. We propose a novel approach (DomainEA) that can be effectively implemented on domain-oriented knowledge graphs. Existing methods are aimed at enhancing cross-lingual links, but when applied to the domain-oriented task of entity alignment, they have difficulty effectively solving the aforementioned problems in the introduction. In response to the issue that domain KGs have few relationships between entities, our approach integrates attribute triples instead of relationship triples to achieve the task of entity alignment. Specifically, we first employed a filtering module consisting of attribute values to exclude noisy entities, and selected a candidate set for each entity to reconstruct the sub-KGs. Then, we employed a SA and an AA to integrate the implicit features of the entities, and to map the heterogeneous KGs onto a unified space. Finally, the alignment module realized the alignment task by calculating the similarity among the entity embeddings. Compared to the five baselines, our approach consistently performed better on real datasets, and comparisons from the ablation experiments also demonstrate the usefulness of the filtering module and SA.

DomainEA achieves some advancements in the task of domain-oriented entity alignment. However, it also has some limitations to its applications. We will pay more attention to investigating two aspects of our approach in future work:

Firstly, DomainEA only performed well on two Chinese-recipe datasets, which limits its universality. So we will evaluate DomainEA on multilingual datasets, e.g., English and French, and on domain-oriented datasets, e.g., biomedical science and social media.

Secondly, we utilized attribute values to capture the features of entities, but entities also have other useful features, such as entity descriptions and entity names. Therefore, we will enhance DomainEA by combining these characteristics in future work.

Author Contributions: Conceptualization, J.Z.; methodology, J.Z. and Y.X.; software, Y.G., H.L. and Y.L.; validation, P.L.; data curation, C.L., Y.L., H.L. and Y.Z.; writing—original draft preparation, Y.X.; writing—review and editing, Y.L., J.Z. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62102372, 62072414, and 62006212; The Program for Young Key Teachers of Henan Province under Grant No. 2021GGJS095; The Project of Science and Technology in Henan Province under Grant Nos: 222102210317, 232102210078, 232102240072, 232102210023, and 232102211051; The Project of Collaborative Innovation in Zhengzhou under Grant Nos. 2021ZDPY0208; The Doctor Scientific Research Fund of Zhengzhou University of Light Industry under Grant No. 2021BSJJ029.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Notably, the dataset has been released on GitHub <https://github.com/ZhongJinjun/DomainEA> (accessed on 31 December 2022) to facilitate access for scholars who may wish to employ it for research purposes.

Acknowledgments: We sincerely thank the editors and reviewers for their valuable comments in improving this paper.

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

References

1. Zeng, K.; Li, C.; Hou, L.; Li, J.; Feng, L. A comprehensive survey of entity alignment for knowledge graphs. *AI Open* **2021**, *2*, 1–13. [[CrossRef](#)]
2. Shen, L.; He, R.; Huang, S. Entity alignment with adaptive margin learning knowledge graph embedding. *Data Knowl. Eng.* **2022**, *139*, 101987. [[CrossRef](#)]
3. Huang, H.; Li, C.; Peng, X.; He, L.; Guo, S.; Peng, H.; Wang, L.; Li, J. Cross-knowledge-graph entity alignment via relation prediction. *Knowl. Based Syst.* **2022**, *240*, 107813. [[CrossRef](#)]
4. Xu, Y.; Li, Z.; Chen, Q.; Wang, Y.; Fan, F. An Approach for Reconciling Inconsistent Pairs Based on Factor Graph. *J. Comput. Res. Dev.* **2020**, *57*, 175–187.
5. Huang, J.; Wang, J.; Li, Y.; Zhao, W. A Survey of Entity Alignment of Knowledge Graph Based on Embedded Representation. *J. Phys. Conf. Ser.* **2022**, *2171*, 012050. [[CrossRef](#)]
6. Xu, Y.; Li, Z.; Chen, Q.; Fan, F. GL-RF: A reconciliation framework for label-free entity resolution. *Front. Comput. Sci.* **2018**, *12*, 1035–1037. [[CrossRef](#)]
7. Weishan, C.; Yizhao, W.; Shun, M.; Jieyu, Z.; Yuncheng, J. Multi-heterogeneous neighborhood-aware for Knowledge Graphs alignment. *Inf. Process. Manag.* **2022**, *59*, 102790.
8. Usman, A.M.; Liu, J.; Xie, Z.; Liu, X.; Sheeraz, A.; Huang, B. Entity alignment based on relational semantics augmentation for multilingual knowledge graphs. *Knowl. Based Syst.* **2022**, *252*, 109494.
9. Chen, L.; Tian, X.; Tang, X.; Cui, J. Multi-information embedding based entity alignment. *Appl. Intell.* **2021**, *51*, 8896–8912. [[CrossRef](#)]
10. Liu, J.; Chai, B.; Shang, Z. A cross-lingual medical knowledge graph entity alignment algorithm based on neural tensor network. *Basic Clin. Pharmacol. Toxicol.* **2021**, *128*, 31–32.
11. Zhu, B.; Bao, T.; Liu, L.; Han, J.; Wang, J.; Peng, T. Cross-lingual knowledge graph entity alignment based on relation awareness and attribute involvement. *Appl. Intell.* **2023**, *53*, 6159–6177. [[CrossRef](#)]
12. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
13. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
14. Yao, X.; Xie, Y. An Improved Mapping Method of Comprehensive Ontology Similarity. *Comput. Mod.* **2014**, *61*–65.
15. Suchanek, F.M.; Serge, A.; Pierre, S. PARIS: Probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.* **2011**, *5*, 157–168. [[CrossRef](#)]
16. Shao, C.; Hu, L.; Li, J.; Wang, Z.; Chung, T.; Xia, J. RiMOM-IM: A Novel Iterative Framework for Instance Matching. *J. Comput. Sci. Technol.* **2016**, *31*, 185–197. [[CrossRef](#)]
17. Li, Y.; Gao, D. Research on Entities Similarity Calculation in Knowledge Graph. *J. Chin. Inf. Process.* **2017**, *31*, 140–146.
18. Cohen, W.W.; Richman, J. Learning to match and cluster large high-dimensional data sets for data integration. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Montreal, QC, Canada, 23–26 July 2002; pp. 475–480.
19. Verykios, V.S.; Moustakides, G.V.; Elfekey, M.G. A Bayesian decision model for cost optimal record matching. *VLDB J.* **2003**, *12*, 28–40. [[CrossRef](#)]

20. Li, L. Research on Entity Alignment Method for Linked Open Data. Master's Thesis, Beijing University of Chemical Technology, Beijing, China, 2017.
21. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
22. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1544–1550.
23. Xiao, H.; Huang, M.; Hao, Y.; Zhu, X. TransG: A generative mixture model for knowledge graph embedding. *arXiv* **2015**, arXiv:1509.05488.
24. Lin, Y.; Liu, Z.; Sun, M. Modeling Relation Paths for Representation Learning of Knowledge Bases. *arXiv* **2015**, arXiv:1506.00379.
25. Huang, W.; Li, G.; Jin, Z. Improved knowledge base completion by the path-augmented TransR model. In Proceedings of the Knowledge Science, Engineering and Management: 10th International Conference, Melbourne, VIC, Australia, 19–20 August 2017; pp. 149–159.
26. Cao, Y.; Liu, Z.; Li, C.; Liu, Z.; Li, J.; Chua, T.-S. Multi-Channel Graph Neural Network for Entity Alignment. *arXiv* **2019**, arXiv:1908.09898.
27. Wu, Y.; Liu, X.; Feng, Y.; Wang, Z.; Zhao, D. Jointly Learning Entity and Relation Representations for Entity Alignment. *arXiv* **2019**, arXiv:1909.09317.
28. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the The Semantic Web: 15th International Conference, Heraklion, Greece, 3–7 June 2018; pp. 593–607.
29. Wang, C.; Huang, Z.; Wan, Y.; Wei, J.; Zhao, J.; Wang, P. FuAlign: Cross-lingual entity alignment via multi-view representation learning of fused knowledge graphs. *Inf. Fusion* **2023**, *89*, 41–52. [[CrossRef](#)]
30. Teong, K.-S.; Soon, L.-K.; Su, T.T. Schema-agnostic entity matching using pre-trained language models. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Birmingham, UK, 19–23 October 2020; pp. 2241–2244.
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. Liu, Z.; Cao, Y.; Pan, L.; Li, J.; Chua, T.-S. Exploring and evaluating attributes, values, and structures for entity alignment. *arXiv* **2020**, arXiv:2010.03249.
33. Thirumuruganathan, S.; Li, H.; Tang, N.; Ouzzani, M.; Govind, Y.; Paulsen, D.; Fung, G.; Doan, A. Deep learning for blocking in entity matching: A design space exploration. *Proc. VLDB Endow.* **2021**, *14*, 2459–2472. [[CrossRef](#)]
34. Nie, H.; Han, X.; Sun, L.; Wong, C.; Chen, Q.; Wu, S.; Zhang, W. Global structure and local semantics-preserved embeddings for entity alignment. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 3658–3664.
35. Xiang, Y.; Zhang, Z.; Chen, J.; Chen, X.; Lin, Z.; Zheng, Y. OntoEA: Ontology-guided entity alignment via joint knowledge graph embedding. *arXiv* **2021**, arXiv:2105.07688.
36. Liu, F.; Vulić, I.; Korhonen, A.; Collier, N. Learning domain-specialised representations for cross-lingual biomedical entity linking. *arXiv* **2021**, arXiv:2105.14398.
37. Azzalini, F.; Jin, S.; Renzi, M.; Tanca, L. Blocking Techniques for Entity Linkage: A Semantics-Based Approach. *Data Sci. Eng.* **2021**, *6*, 20–38. [[CrossRef](#)]
38. Muhammad, E.; Saravanan, T.; Shafiq, J.; Mourad, O.; Nan, T. Distributed representations of tuples for entity resolution. *Proc. VLDB Endow.* **2018**, *11*, 1454–1467.
39. Javdani, D.; Rahmani, H.; Allahgholi, M.; Karimkhani, F. Deepblock: A novel blocking approach for entity resolution using deep learning. In Proceedings of the 2019 5th International Conference on Web Research (ICWR), Cambridge, UK, 26–28 August 2019; pp. 41–44.
40. Zhang, W.; Wei, H.; Sisman, B.; Dong, X.L.; Faloutsos, C.; Page, D. Autoblock: A hands-off blocking framework for entity matching. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 744–752.
41. Tang, J.; Zhao, K.; Li, J. A Fused Gromov-Wasserstein Framework for Unsupervised Knowledge Graph Entity Alignment. *arXiv* **2023**, arXiv:2305.06574.
42. Tang, W.; Su, F.; Sun, H.; Qi, Q.; Wang, J.; Tao, S.; Hao, Y. Weakly Supervised Entity Alignment with Positional Inspiration. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 814–822.
43. Zhong, Z.; Zhang, M.; Fan, J.; Dou, C. Semantics driven embedding learning for effective entity alignment. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 2127–2140.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.