electronics

MDPI

Article

Fusing Attribute Character Embeddings with Truncated Negative Sampling for Entity Alignment

Hongchan Li, Zhuang Zhu, Haodong Zhu * and Baohua Jin

Department of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
* Correspondence: zhdzzuliketizu@163.com

Abstract: Finding pairs of entities from two different knowledge graphs that reflect the same real-world object is the purpose of entity alignment for knowledge graphs. In recent years, techniques that use entity alignment for knowledge fusion have received widespread attention. In this article, we suggest a method for entity alignment using truncated negative sampling with attribute character embedding. The method mainly makes use of the relationship and attribute data in heterogeneous knowledge graphs to fulfil the entity alignment task. Firstly, the framework uses relationship mapping to unify the namespace of heterogeneous relationships. Secondly, the attribute character embeddings are generated using the attribute triples in the knowledge graph to unify the embedding space of heterogeneous entities. Then, the entity similarity between heterogeneous knowledge graphs is captured by structural embedding. Next, to learn more useful semantic information during negative sampling, the framework adopts a truncated negative sampling strategy to increase the generalizability of the model. The negative sampling procedure employs targets with high similarity to the target entity as negative sample targets. Finally, we performed comparison tests on two well-known real-world datasets, and the outcomes demonstrate that the proposed model outperforms three other representative advanced approaches, especially with an over 10% improvement in the Hits@k metric compared to the baseline method.

Keywords: knowledge graph; entity alignment; representation learning; character embedding

Citation: Li, H.; Zhu, Z.; Zhu, H.; Jin, B. Fusing Attribute Character Embeddings with Truncated Negative Sampling for Entity Alignment. *Electronics* **2023**, *12*, 1947. https://doi.org/10.3390/electronics12081947

Academic Editor: Cheng-Chi Lee

Received: 2 April 2023
Revised: 17 April 2023
Accepted: 19 April 2023
Published: 21 April 2023

1. Introduction

Knowledge graphs, which store human knowledge in a structured way, play an increasingly important role in artificial intelligence and natural language processing. Typical KGs represent a piece of knowledge in the form of a triple, namely (head entity, relation, tail entity) or (h, r, t) [1]. Large-scale KGs have developed rapidly in recent years, such as Freebase [2], DBpedia [3], Wikidata [4], Probase [5], YAGO [6–8], and NELL [9]. These KGs, which are widely utilized in applications such as knowledge question answering and knowledge reasoning, provide structural information about entities and relations in the world.

However, most of the existing knowledge graphs are developed separately for different needs and purposes, which inevitably results in heterogeneity in the data of these knowledge graphs, and their content is also complementary. Integrating heterogeneous and complementary knowledge into a more powerful and comprehensive knowledge base is an urgent and feasible task. To effectively support knowledge-driven applications, entity alignment consolidates complementary knowledge from disparate knowledge graphs.

Identifying pairs of entities in heterogeneous knowledge graphs that represent the same real-world objects is the key to entity alignment. In this paper, we focus on entity alignment between two knowledge graphs, and most of the research is based on storing entity data in the form of RDF triples. An RDF triple is composed of three elements, subject, predicate/relation, and object, where the subject represents an entity and the object can

be either an entity or a character. The triple is labelled as a relation triple if the object is an entity, and an attribute triple if the object is a character. Figure 1 shows RDF triples for two knowledge graphs, G1 and G2 (the prefixes "Wiki" and "DBP" are simplified original URIs). Entity 'Wiki: Q36687' and entity 'DBP: Victoria' are aligned entities in heterogeneous knowledge graphs, and relationships can be unified through relation mapping. Then, the complementary knowledge of heterogeneous entities can be fused, as shown in G1_2 in the figure.

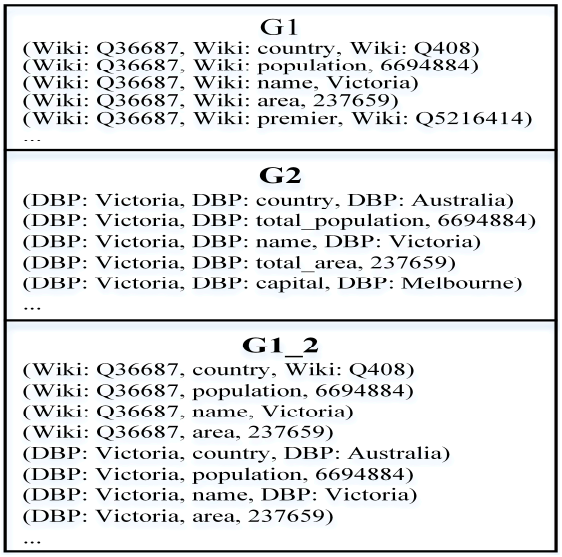

Figure 1. Example of entity alignment.

The original studies of entity alignment relied on similarities in the attributes of the entities. These techniques mainly rely on manually created comparison criteria. However, the same entity in different heterogeneous knowledge graphs may have different property names, and different entities may require a comparison of different entity properties. Therefore, methods relying on manually defined rules are prone to errors and difficult to implement.

Recent years have seen the development of embedding-based entity alignment models, such as TransE [1]. This model learns entity embeddings by using relationship triples in the knowledge graph to measure the knowledge graph's elements' semantic similarity. Embedding-based models require the embeddings from two graphs to fall into the same vector space for KG embeddings to be suitable for entity alignment between two graphs. To solve this problem, MTransE [10] encodes entities and relationships separately in their own embedding space and provides a transformation matrix for vector space conversion, but this method may result in information loss during space transformation. The validity and quantity of seed entity pairs is critical to these techniques, and the very limited number of seed entities currently available in heterogeneous knowledge graphs and their quality is difficult to guarantee, requiring significant manual effort to obtain.

As far as we know, there are numerous attribute triples in the knowledge graph that contain potential semantic information. This latent semantic information can better model the knowledge graph and is useful for capturing similarities. Existing approaches do not make good use of attribute triples. This paper's approach uses attribute triples to generate attribute character embeddings to mine potential semantic information, which can better capture similarity.

This paper suggests embedding two knowledge graphs into a single vector space by learning the attribute embeddings from the attribute triples. Entity embeddings can capture the entity similarities between two knowledge graphs since the similarities between characteristics in the two graphs can create a single vector space for the two graphs. To

ensure that the relationships in the two graphs may be embedded into a single vector space, the model contains a relation mapping module that renames relationships in two graphs using a uniform naming strategy.

This paper makes many contributions, such as (1) proposing an entity alignment method that includes relation mapping, embedding learning, and entity alignment; (2) solving the issue of inconsistent vector space by using entity attribute character embeddings; and (3) improving generalization ability through the use of truncated negative sampling.

2. Related Work

2.1. Entity Alignment Based on Semantic Matching

By comparing the latent semantics of entities with the relationships reflected in vector space representations, semantic matching models utilize similarity-based scoring functions to gauge the rationality of facts. RESCAL [11] was the earliest semantic matching model, and its representation learning process is generally completed through tensor decomposition. RESCAL models the relationship (r) as a matrix (Mr) to capture pairwise interactions between latent factors of entities. SimplE [12] restricts the relationship matrix (Mr) to a diagonal matrix and emphasizes that the roles played by the entity (e), as the subject and object in the triple should be distinguished. SimplE encodes two embedding vectors, e_h and e_t, for each entity (e), corresponding to the head entity and tail entity used in the entity triple, respectively. CrossE [13] believes that bidirectional effects between entities and relationships help select relevant information in link prediction; therefore, in addition to learning universal embeddings for each entity and relationship, CrossE also learns additional embedding (c_r) for each relationship (r) to model bidirectional interactions between entities and relationships. According to RotatE [14], which can model and infer different relationship patterns, relationships are described as rotation changes in complex space. MuRP [15] points out that the relationships between entities have a hierarchy and modelled entities in hyperbolic space. HAKE [16] believes that polar coordinates are more suitable for representing the semantic hierarchy of entities and represent each entity as a modulus and phase. Entity alignment, which is based on semantic matching, models the entities and relationships in a knowledge graph by learning its structural information, but the entity attributes, which contain a significant amount of semantic information, have not been fully exploited.

2.2. Entity Alignment Based on Representation Learning

In recent years, research on knowledge graph link prediction has improved considerably in the use of knowledge graph embedding techniques. The aim of this kind of operation is prediction, discovering entities and relationships that may exist through existing entities and relationships in the knowledge graph, thereby expanding the knowledge base. Among the existing methods, TransE [1] performs well in link prediction, which translates the relationship between the head entity and tail entity into the transformation of entity embeddings. In recent years, to address the limitations of the TransE model in handling complex relationships, researchers have proposed improved models. The head entity and tail entity are projected into the relationship space using two different projection matrices by the TransD [17] model, respectively. According to TransR [18], each entity is made up of several attributes, and various relationships concentrate on a distinct attribute of an entity while having their own semantic space. The TransR model entities and relationships in two different spaces, i.e., the entity space and multiple relationship spaces (relationship-specific entity space), and performs a transformation in the corresponding relationship space. TransH [19] thinks an entity can have many representations depending on the relationships it is involved in. For the relationship (r), the TransH model uses both translation vectors and hyperplane normal vectors to represent it.

Entities with comparable neighbour structures in the knowledge graph should have compact representations in the embedding space, as this is the goal of embedding models,

which seek to retain the structural information of entities. This progress in embedding models has led researchers to study entity alignment based on embeddings. IPTransE [20] uses an iterative method of joint embedding for entity alignment. The joint embedding between knowledge graphs is initially calculated using alignment seed entities. Then, the joint embedding is iteratively updated using newly aligned entities. DAT [21] encodes both entity names and entity relationships to jointly represent entity information. Encoding entity names supplements entity name semantics and performs well in handling tail entities. TransEdge [22] interacts with embeddings between head entities and tail entities based on their relationships and represents the relationship as a composite function of the interaction embeddings. MultiKE [23] uses multiple views of entities to embed entities and aligns them using a combination of name, relationship, and attribute views. BootEA [24] proposes iterative entity alignment and attribute correlation strategies and solves the problem of insufficient seed entities by iteratively adding aligned entities to the training set. A cross-lingual entity alignment approach with joint attribute-preserving embedding is proposed by JAPE [25], and entity embeddings are improved by utilizing attribute data that have been abstracted to related data types. KD-CoE [26] jointly learns multi-lingual entity description information for cross-lingual entity alignment. Potential semantic information in the knowledge graph can be captured by using entity attribute data. For entity alignment tasks, this work leverages entity attribute values.

3. Problem Definition

Entity alignment seeks to identify pairings of entities that reflect the same real-world entity in the two heterogeneous knowledge graphs, designated as G1 and G2, respectively. In the heterogeneous knowledge graphs, there exist pre-aligned entity pairs (referred to as seed entities), and the embedding of the seed entities yields the knowledge graphs' structural information. It is expected that potentially aligned entities will have compact vector representations in the vector space.

The building blocks of a knowledge graph are triples with the forms <h, r, t> and <h, r, a>, where h and t stand for entities, r for relationships between entities, and a for attribute values of entities. The entity alignment assignment seeks to identify pairs of knowledge graphs, G1 and G2, with the properties, h1 $\in$ G1, h2 $\in$ G2, where <h1, h2> indicate the same real-world entity. The elements in the triples are represented as low-dimensional vectors in the form of <**h**, **r**, **t**> and <**h**, **r**, **a**> during the entity embedding process, where the vector depiction of entities in the vector space is indicated by bold letters.

Embedding-Based Entity Alignment Method

According to TransE, to preserve the structure information of the knowledge graph during entity embedding, the sum of the embeddings of the head entity (h) and the relationship (r) should be close to the embedding of the tail entity (t) for a given triple of <h, r, t>, i.e., $h + r \approx t$. By this means, entities with comparable neighbour structures have condensed representations in the embedding space. To learn structural embedding, TransE minimizes the following objective function:

$$J_s = \sum_{e_r \in E_r} \sum_{e'_r \in E'_r} max\left(0, \left[\theta + j(e_r) - j\left(e'_r\right)\right]\right) \tag{1}$$

$$E_r = \{\langle h, r, t\rangle | \langle h, r, t\rangle \in G\}\ ; j(e_r) = ||h + r - t|| \tag{2}$$

$$E'_r = \left\{\left\langle h', r, t\right\rangle \middle| h' \in S\right\} \cup \left\{\left\langle h, r, t'\right\rangle \middle| t' \in S\right\} \tag{3}$$

Here, $||v||$ is the L1 normalization of the vector (v); θ is the margin hyperparameter; E_r is the set of valid relationship triples from the training dataset. E'_r is the set of invalid

triples (where *S* is the set of entities in *G*), which replaces the head or tail entity in the triple of positive examples with a random entity.

The characteristics of structural embedding have promoted further research on entity alignment. However, there are significant limitations to directly using structural embeddings. For example, the embedding spaces of heterogeneous knowledge graphs are inconsistent, making it difficult to perform similarity calculations. Although MTransE proposes to compute a spatial transformation matrix to convert embedding vectors in different spaces into the same vector space for similarity calculation, the computation of the transformation matrix requires numerous seed entity pairs. The matrix is also easily influenced by the quality of the seed entities. In addition, embedding-based models have limited modelling capabilities for one-to-many relationships, such as the triple <Curry, born in, Akron> and <James, born in, Akron>, where the embedding model can cause the Curry and James entities' vector representations to be too similar.

4. Model

4.1. Model Overview

The method is made up of three parts, relationship mapping, embedding learning, and entity alignment. The entity alignment process of the model is shown in Figure 2.

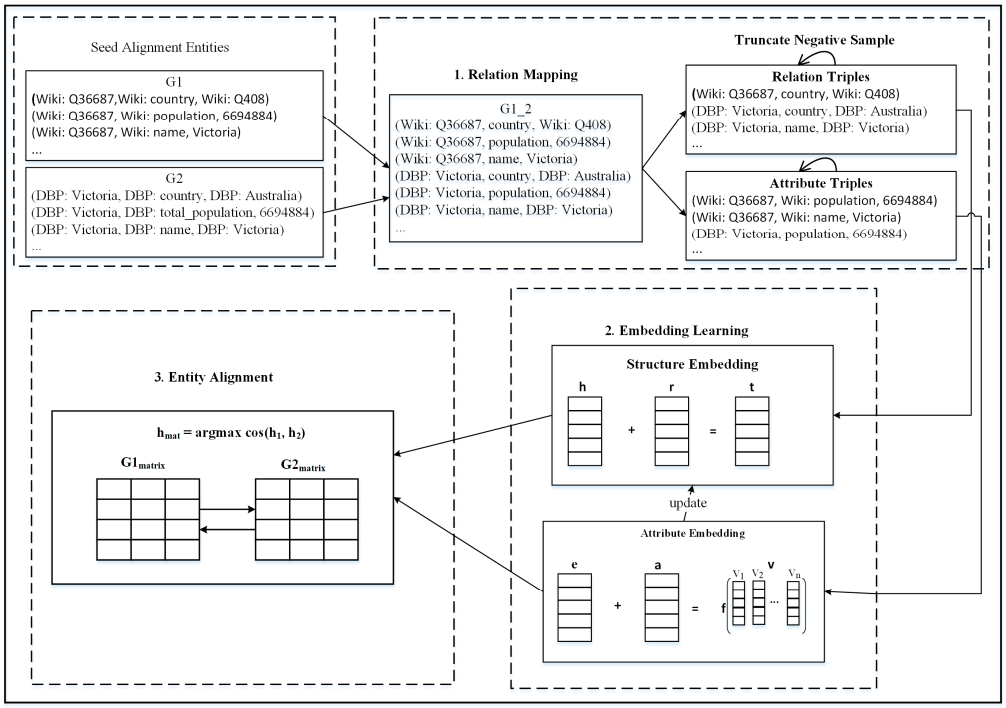

Figure 2. Entity alignment process.

To better measure the similarity of entities, it is necessary to embed entities and relationships into a unified vector space. To offer a single vector space for the knowledge graph's structural embedding, the relationship mapping module (Section 4.2) is used to find similar relationships and name them using a unified naming scheme, such as naming both "locate in" and "be located in" as "locate in". After the relationship mapping, the entity set is obtained by taking the union of G1 and G2, and, depending on whether the tail entity (t) in the triple is an entity or an attribute value, the entity set is separated into relationship triples and attribute triples.

In the embedding learning module, relationship triples and attribute triples are utilized to jointly learn the embeddings of the two knowledge graphs. The relationship embeddings and structural information of the knowledge graph are learned from the relationship triple set, while the attribute embeddings are learned from the attribute triple set, which can

uncover more potential semantic information. The embeddings of relationship triples in G1 and G2 belong to different vector spaces because the entity naming schemes of the two heterogeneous knowledge graphs are not the same. However, the attribute embeddings are based on attribute characters, and the embeddings of attribute triples can fall into a single vector space, so the attribute character embeddings of heterogeneous knowledge graphs can be similar. The structural embeddings of entities can be transformed into a unified vector space using attribute character embeddings, which can also capture the similarity between entities in two diverse knowledge graphs throughout the learning process.

After unifying the vector space, all entity embeddings from the two knowledge graphs can be obtained in the unified space. The entity alignment module will find every pair <h1, h2> with a similarity greater than the set threshold, where h1 ∈ G1 and h2 ∈ G2. To improve the model's generalization ability, this paper uses a truncated negative sampling strategy instead of the traditional random negative sampling strategy. This allows the model to learn more useful information during the learning process.

4.2. Relation Mapping

To embed entities into a unified vector space, the relationship mapping module renames the relationships of the two heterogeneous knowledge graphs using a unified naming scheme. For the relationships that have already been aligned in the seed entities, such as 'G1: "locate in"' and 'G2: "be located in"', the relationship mapping module will rename these aligned relationships using a unified naming scheme (such as "locate in"). To discover more matching relationships during the model learning process, the similarity between two relationships can be calculated using the edit distance, and relationships with a similarity greater than the set threshold (0.9) are considered aligned relationships.

4.3. Embedding Learning

4.3.1. Structure Embedding

We employ the idea of TransE for the structural embedding of knowledge graphs. During the learning process to obtain more information from the aligned relationships, a learning weight (α) is added to control the learning process and focus more on the aligned triples (i.e., triples that contain aligned relationships), thereby achieving entity alignment between knowledge graphs. In the model training, the structural embedding minimizes the following objective function:

$$J_s = \sum_{e_r \in E_r} \sum_{e'_r \in E'_r} max\left(0, \left[\gamma + \alpha\left(j(e_r) - j(e'_r)\right)\right]\right) \tag{4}$$

$$\alpha = \frac{total(r)}{|T|} \tag{5}$$

Here, total (r) is the number of times the relation (r) has occurred; E_r is the set of triples with positive relationships; E'_r is the set of triples with negative relationships; and $|T|$ is the total number of triples in the union of the two knowledge graphs G1 and G2. Since the aligned relationships will appear in both knowledge graphs, their frequency is higher than that of the non-aligned relationships. Adding the α weight to focus more on the aligned relationships during the model learning process will help obtain more structural information. For example, if there are six relationships and properties shared by two aligned entities, 'G1: "locate in"' and 'G2: "be located in"' are aligned relationships, α weight is $2/6$, and the weight of other non-aligned relationships is $1/6$. This can help the model learn more valuable information from the aligned relationships.

4.3.2. Attribute Embedding

When embedding attributes, the relationship (r) can be interpreted as a transformation from the head entity (h) to the attribute character (a). However, for heterogeneous knowledge graphs, the same attribute may have different representations, such as numerical

attribute values 35.32445 and 35.3244989996 or string attribute values "Stephen Curry" and "Wardell Stephen Curry II," among others. For heterogeneous attribute values, a composite function is used for embedding. Elements of the attribute triple (e, a, v) are defined as $e + a \approx f_v\ (v)$, where $f_v\ (v)$ is the composite function used for character embedding v is the attribute value $v = \{u_1, u_1, \ldots, u_i\}$. During the embedding process, the attribute value is embedded as a low-dimensional vector. Similar attribute values have similar vector representations. The two composite functions are described in the following.

Add Function: The add function takes the embedded vectors of each character of the attribute value and adds them together to represent the attribute value. However, this function has certain problems and can introduce incorrect learning samples. For example, for the attribute values "danger" and "garden," although the character order is different, they have the same embedding representation after being processed by the add function.

$$f_v(v) = \boldsymbol{u}_1 + \boldsymbol{u}_2 + \ldots + \boldsymbol{u}_i \tag{6}$$

N-Gram Function: To address the problem of errors introduced by the add function, the N-Gram function [14] is used for character embedding.

$$f_v(v) = \sum_{n=1}^{N} \sum_{i=1}^{t} \frac{\sum_{j=i}^{n} u_j}{t - i - 1} \tag{7}$$

Here, N represents the maximum value of N used in the n-gram combination and t is the length of the attribute value. The objective function for attribute embedding is:

$$J_{CE} = \sum_{t_v \in T_v} \sum_{t'_v \in T'_v} \max\left(0, \left[\eta + \alpha\left(f(t_v) - f\left(t'_v\right)\right)\right]\right) \tag{8}$$

$$T_v = \{\langle e, a, v\rangle \in G\}; f(t_v) = ||\boldsymbol{h} + \boldsymbol{r} - f_v(v)|| \tag{9}$$

$$T'_v = \left\{\left\langle e', a, v\right\rangle \middle| e' \in E\right\} \cup \left\{\left\langle e, a, v'\right\rangle \middle| v' \in A\right\} \tag{10}$$

In the formula, T_v is the set of positive attribute triples; T'_v is the set of negative triples (where A is the set of attribute values in the knowledge graph G). The negative attribute triples are generated using a truncated negative sampling strategy. $f_v(v)$ is the confidence score of the embedding head entity (e), relationship (a), and attribute value (v) calculated using the N-Gram composite function.

4.4. Truncated Negative Sampling

In entity alignment research, most models use random negative sampling, which replaces the head or tail entity of a positive triple (h, r, t) with another entity at random to generate negative samples. However, the strategy of random negative sampling has limited the help for model learning. For example, for the triple <Curry, born in, Akron>, random negative sampling may replace the tail entity with James, which generates negative samples that are wrong and meaningless. This is because the replaced entity may be orthogonal to the original entity in the vector space. In contrast, truncated negative sampling seeks entities that are highly similar to the replaced entity as negative samples, improving the model's recognition ability and allowing more information to be learned from negative samples. Specifically, for a given entity (e) to be replaced, a set of entities similar to it is searched for from the embedding space as negative sampling targets, i.e., num = $\lceil(1 - \mu)N\rceil$, where $\mu \in [0,1)$ is the sampling ratio, N is the number of entities in the knowledge graph, and $\lceil . \rceil$ is the ceiling function. Therefore, entities with low similarity to entity (e) are truncated and not sampled.

$$\mathrm{SIM} = \sum_{e \in G1 \cup G2} [1 - \mathrm{sim}(e_{se}, e_{ce})] \tag{11}$$

(e_{se}, e_{ce}) represents the similarity between the structural embedding vector and attribute embedding vector of entity (e), and cosine similarity is used as the similarity measure in the model. Structural embedding learns the structural information of two knowledge graphs through the relationships between entities, while attribute embedding can discover potential semantic information. The following is the objective function for learning structural embedding and attribute embedding jointly:

$$J = J_S + J_{CE} + SIM \tag{12}$$

4.5. Entity Alignment

The entities of knowledge graph G1 and G2 are embedded into a single vector space by the joint learning of attribute embedding and structure embedding, and similar entities have similar vector embeddings. The following is the entity alignment calculation formula:

$$h_{mat} = argmax\ \cos(h_1, h_2) \tag{13}$$

Here, $h_1 \in G_1$ and $h_2 \in G_2$. Given h_1, we calculate the similarity between all entities in G_2 and h_1, and put the entities with similarity greater than the set threshold into the h_{mat} set.

5. Experiment

5.1. Experiment Settings

Environment information: The experiment was conducted on a personal computer equipped with an AMD Ryzen 7 4800H 2.9 GHz CPU, NVIDIA GeForce GTX 1650 Ti GPU, and 16 GB RAM.

Dataset: To confirm the model's efficacy on real-world knowledge graph, the model was evaluated on Dbpedia, Wikidata, and YAGO3 real-world knowledge graph data constructed by OpenEA [27]. Table 1 presents the data statistics of the dataset D_W_15K, which contains 15,000 seed entities. Table 2 presents the cross-lingual entity alignment dataset EN-DE-15K (English–German). The entities in both datasets primarily include writers, cities, music, actors, etc. For both datasets, statistical data are provided in Tables 1 and 2, including the total number of relations, attributes, relationship triples, and attribute triples.

Table 1. Data statistics of the monolingual datasets.

Dataset		Relations	Attributes	Relation Triples	Attribute Triples
D-W15K	DB	248	342	38,265	68,258
	WD	169	649	42,746	138,246

Table 2. Data statistics of the cross-lingual datasets.

Dataset		Relations	Attributes	Relation Triples	Attribute Triples
EN-DE15K	EN	215	286	47,676	83,755
	DE	131	194	50,419	156,150

Implementation details: To compare performance, four embedding-based entity alignment methods were selected for performance comparison. The experimental details are detailed in the following. For the TransE model, its complete entity is used for embedding. The MTransE paper implemented five variants and, according to the performance of variant four in the paper, variant four of MTransE is used as a comparison. JAPE uses an abstract data type for entity attributes, and its full model is used in the experiment.

IPTransE provides three variant models, including a translation-based model, a linear transformation-based model, and a parameter-sharing-based model. The experiments show that the parameter-sharing-based model has the best performance. Therefore, in the experiments, we used the parameter-sharing-based model of IPTransE. For the model in this paper, the configurations used were γ = 1.5 and μ = 0.95, and 5 negative samples were truncated for each entity. The parameter settings of the compared methods follow the best configuration in the original paper. To accurately compare each model's entity alignment performance, the embedding dimension is unified to 100, the maximum epoch is 3000, the batch size is 3000, and the learning rate is 0.01.

Evaluation metrics: In the experiment, hits@k (k = 1, 5, 10, 50), MRR, and MR are used as evaluation metrics to evaluate the performance of the models. Among them, hits@k indicates the proportion of correctly aligned entities among the top k predicted correctly aligned entities during the entity alignment process. MRR is the average value of the reciprocal of the ranking of correctly aligned entities, and MR is the average ranking of correctly aligned entities. These are used to gauge the model's effectiveness and the percentage of successfully aligned entities in the top K predictions. The model performs better when the hits@k and MRR are bigger and the MR is smaller.

5.2. Experimental Result and Analysis

Tables 3 and 4 show the performance of different models on monolingual and cross-lingual datasets, where the hits@k metric represents the percentage (%) and bold numbers represent the performance of the proposed model.

Table 3. D-W15K entity alignment results.

Model	Hits@1	Hits@5	Hits@10	Hits@50	MRR	MR
TransE	8.35	17.33	28.57	35.76	0.13	22007
MtransE	16.51	32.47	39.51	57.84	0.24	412
JAPE	14.45	29.44	36.92	56.88	0.22	258
IPTransE	32.73	43.56	54.37	69.38	0.37	240
My Model	**32.41**	**49.02**	**56.16**	**79.92**	**0.41**	**235**

Bold numbers indicate the performance indicators of the method in this paper.

Table 4. EN-DE15K entity alignment results.

Model	Hits@1	Hits@5	Hits@10	Hits@50	MRR	MR
TransE	6.42	12.57	19.63	23.15	0.11	32107
MtransE	11.90	24.61	31.25	47.20	0.18	565
JAPE	12.24	26.26	34.38	51.95	0.20	357
IPTransE	35.22	51.36	79.52	88.27	0.68	117
My Model	**65.90**	**79.52**	**83.87**	**91.67**	**0.72**	**48**

Bold numbers indicate the performance indicators of the method in this paper.

The proposed model uses Equation (12) to calculate the similarity between heterogeneous knowledge graph entities. It can be seen from Tables 3 and 4 that the proposed model has different degrees of improvement in Hits@k, MRR, and MR compared with the baseline approach. In particular, the improvement in the Hits@k metric is significant compared with the best baseline model, which indicates that the accuracy of the method in this paper is good in terms of entity alignment.

The performance of the TransE model is the worst among the baseline models, both on the D-W15K and EN-DE15K datasets, because TransE uses only relational triples to model the structure of the knowledge graph and not attribute triples rich in latent semantics. In

addition, the inconsistency of the embedding space of heterogeneous knowledge graphs causes TransE to have difficulty capturing the knowledge graphs during the learning process. The performance of MtransE is improved on two datasets compared to TransE, but the performance on cross-lingual datasets is average. Because the heterogeneity of the cross-linguistic data is more pronounced when learning the spatial transformation matrix during embedding learning, MTransE performs relatively poorly on cross-linguistic datasets. JAPE uses attribute triples containing potential semantic information to model the knowledge graph, but the attribute values are converted into corresponding data types for embedding learning, which distorts the original semantics of part of the attribute triples. The MRR metrics of IPTransE are comparable to our method for both monolingual and cross-language datasets. This is mainly because IPTransE models the multi-step relational paths and captures the semantic similarity between entities better. In addition, during iterative training, IPTransE adds aligned and plausible entity pairs to the training data via parameter sharing, thus enriching the seed entity set and helping to learn the structural similarity of the knowledge graph better. However, our approach outperforms IPTransE in terms of Hits@1 metrics, especially on cross-lingual datasets. This is because our approach uses attribute character embeddings rich in semantic information and better models the structure of the knowledge graph using a truncated negative sampling strategy. In addition, Table 4 shows that the proposed method has a significant improvement in Hits@1 metrics on the cross-lingual dataset compared to the baseline method. This is because the proposed method integrates attribute character embedding into the model, which better models the structure of the knowledge graph and can more accurately find entities that match the target entities. In addition, the truncated negative sampling strategy used in the proposed model is different from the random negative sampling strategy used in the comparison model. The proposed strategy can obtain more valuable information through the negative sampling process and, thus, has a better generalization capability.

In line with expectations, the TransE-based approach performs poorly in the entity alignment task because the embeddings of heterogeneous knowledge graphs fall in different vector spaces. MTransE uses seed entity embeddings to compute transitions between different vector spaces, which are vulnerable to the number and quality of seed entities, and information loss may occur during the transitions. JAPE abstracts attribute information into the corresponding data types, which cannot capture semantic similarity at the character level. IPTransE performs best in the baseline model due to the use of iterative and parameter-sharing strategies. In the proposed model, attribute character embedding can better preserve the similarity between attribute characters and can improve the performance of the model by using more attribute information in the entity alignment process.

6. Conclusions

The translation-based method embeds two heterogeneous knowledge graphs into different vector spaces and uses a transformation matrix to unify the vector spaces for entity alignment. However, the transformation matrix is easily affected by the number and quality of seed entities, and information loss may occur during the space transformation process. This study suggests a knowledge graph entity alignment model that simultaneously learns structural and attribute embeddings and uses attribute character embeddings to unify the vector spaces of structural embeddings. This solves the problems of inconsistent vector spaces and information loss during space transformation. Moreover, the proposed model uses a truncated negative sampling strategy to sample more valuable semantic information during negative sampling in the training process. The joint learning mechanism unifies entity embeddings into a consistent vector space, which is beneficial for similarity calculation. The suggested strategy outperforms baseline models in practical knowledge graph entity alignment tasks, according to experimental findings. Although the performance of our proposed method in terms of entity alignment has been improved compared to the comparison method, there are still two limitations. Firstly, the dependence on the seed entity is still obvious. If the number of seed entities is reduced, the performance of the

method in this paper will be affected. Secondly, the embedding of attribute characters may be influenced by heterogeneity, making it difficult to capture attribute similarity. This is also the direction of our future efforts. In the next step, we will study how to use unsupervised methods for entity alignment to solve the problem of dependence on seed entities. Future work will continue to explore how to discover entity alignment problems in heterogeneous knowledge graphs to improve entity alignment performance.

Author Contributions: Conceptualization H.L. and B.J.; Methodology H.L. and Z.Z.; Software, Z.Z.; Validation, Z.Z.; Resources, H.Z.; Writing—original draft preparation, Z.Z.; Writing—review and editing H.L. and B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Project of Science and Technology Tackling Key Problems in Henan Province of China under Grant 222102210234 and 232102210035.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions, which have improved the presentation.

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

References

1. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *NIPS* **2013**, *26*, 2287–2795.
2. Bollacker, K.; Evans, C.; Sturge, T. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the International Conference on Management of Data, Vancouver, BC, Canada, 10 June 2008.
3. Mendes, P.N.; Jakob, M.; Bizer, C. *DBpedia: A Multilingual Cross-Domain Knowledge Base*; LREC: Istanbul, Turkey, 2012.
4. Vrandečić, D.; Krötzsch, K. Wikidata: A free collaborative knowledgebase. *CACM* **2014**, *57*, 77–85. [CrossRef]
5. Wu, W.; Li, H.; Wang, H.; Zhu, K.Q. Probase: A probabilistic taxonomy for text understanding. In Proceedings of the International Conference on Management of Data, Scottsdale, AZ, USA, 20 May 2012.
6. Hoffart, J.; Suchanek, F.M.; Berberich, K.; Weikum, G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* **2013**, *194*, 28–61. [CrossRef]
7. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the International Conference on World Wide Web, Banff, AB, Canada, 8 May 2007.
8. Mahdisoltani, F.; Biega, J.; Suchanek, F. *YAGO3: A Knowledge Base from Multilingual Wikipedias*; CIDR: Asilomar, CA, USA, 2015.
9. Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. Never-ending learning. *CACM* **2018**, *61*, 103–115. [CrossRef]
10. Chen, M.; Tian, Y.; Yang, M.; Zaniolo, C. *Multilingual Knowledge Graph Embeddings for Cross-Lingual Knowledge Alignment*; IJCAI: Melbourne, Australia, 2017.
11. Nickel, M.; Tresp, V.; Kriegel, H.P. *A Three-Way Model for Collective Learning on Multi-Relational Data*; ICML: Bellevue, WA, USA, 28 June 2011.
12. Kazemi, S.M.; Poole, D. *Simple Embedding for Link Prediction in Knowledge Graphs*; NIPS: Montréal, QC, Canada, 2018.
13. Zhang, W.; Paudel, B.; Zhang, W.; Bernstein, A.; Chen, H. *Interaction Embeddings for Prediction and Explanation in Knowledge Graphs*; WSDM: Melbourne, VIC, Australia, 2019.
14. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. *Rotate: Knowledge Graph Embedding by Relational Rotation in Complex Space*; ICLR: Orleans, LA, USA, 2019.
15. Balazevic, I.; Allen, C.; Hospedales, T. *Multi-Relational Poincaré Graph Embeddings*; NeurIPS: Vancouver, BC, Canada, 2019.
16. Zhang, Z.; Cai, J.; Zhang, Y.; Wang, J. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7 February 2020.
17. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In Proceedings of the Meeting of the Association for Computational Linguistics the International Joint Conference on Natural Language Processing, Beijing, China, 26 July 2015.
18. Moon, C.; Jones, P.; Samatova, N.F. *Learning Entity Type Embeddings for Knowledge Graph Completion*; CIKM: Singapore, 2017.
19. 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 July 2014.
20. Zhu, H.; Xie, R.; Liu, Z.; Sun, M. *Iterative Entity Alignment via Joint Knowledge Embeddings*; IJCAI: Melbourne, VIC, Australia, 2017.
21. Zeng, W.; Zhao, X.; Wang, W.; Tang, J.; Tan, Z. Degree-Aware Alignment for Entities in Tail. In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25 July 2020.
22. Sun, Z.; Huang, J.; Hu, W.; Chen, M.; Guo, L.; Qu, Y. *Transedge: Translating Relation-Contextualized Embeddings for Knowledge Graphs*; ISWC: Auckland, New Zealand, 2019.
23. Zhang, Q.; Sun, Z.; Hu, W.; Chen, M.; Guo, L.; Qu, Y. *Multi-View Knowledge Graph Embedding for Entity Alignment*; IJCAI: Macao, China, 10 August 2019.

24. Sun, Z.; Hu, W.; Zhang, Q.; Qu, Y. *Bootstrapping Entity Alignment with Knowledge Graph Embedding*; IJCAI: Stockholm, Sweden, 2018.
25. Sun, Z.; Hu, W.; Li, C. *Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding*; ISWC: Vienna, Austria, 2017.
26. Chen, M.; Tian, Y.; Chang, K.W.; Skiena, S.; Zaniolo, C. *Co-Training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-Lingual Entity Alignment*; IJCAI: Stockholm, Sweden, 13 July 2018.
27. Sun, Z.; Zhang, Q.; Hu, W.; Wang, C.; Chen, M.; Akrami, F.; Li, C. A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs. *VLDB Endow.* **2020**, *13*, 2326–2340. [CrossRef]