A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs

Zequn Sun [†], Qingheng Zhang [†], Wei Hu [†*], Chengming Wang [†],
Muhao Chen [‡], Farahnaz Akrami [§], Chengkai Li [§]
[†] State Key Laboratory for Novel Software Technology, Nanjing University, China
[‡] Department of Computer Science, University of California, Los Angeles, USA
[§] Department of Computer Science and Engineering, University of Texas at Arlington, USA
{zqsun, qhzhang, cmwang}.nju@gmail.com, whu@nju.edu.cn,
muhaochen@ucla.edu, farahnaz.akrami@mavs.uta.edu, cli@uta.edu

ABSTRACT

Entity alignment seeks to find entities in different knowledge graphs (KGs) that refer to the same real-world object. Recent advancement in KG embedding impels the advent of embedding-based entity alignment, which encodes entities in a continuous embedding space and measures entity similarities based on the learned embeddings. In this paper, we conduct a comprehensive experimental study of this emerging field. We survey 23 recent embedding-based entity alignment approaches and categorize them based on their techniques and characteristics. We also propose a new KG sampling algorithm, with which we generate a set of dedicated benchmark datasets with various heterogeneity and distributions for a realistic evaluation. We develop an open-source library including 12 representative embedding-based entity alignment approaches, and extensively evaluate these approaches, to understand their strengths and limitations. Additionally, for several directions that have not been explored in current approaches, we perform exploratory experiments and report our preliminary findings for future studies. The benchmark datasets, open-source library and experimental results are all accessible online and will be duly maintained.

PVLDB Reference Format:
Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, Chengkai Li. A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs. *PVLDB*, 13(11): 2326-2340, 2020.
DOI: https://doi.org/10.14778/3407790.3407828

1. INTRODUCTION

Knowledge graphs (KGs) store facts as triples in the form of (*subject entity, relation, object entity*) or (*subject entity, attribute, literal value*). This type of knowledge bases supports a variety of applications, e.g., semantic search, question answering and recommender systems [18]. To promote knowledge fusion, researchers have made considerable progress on the task of *entity alignment*, which is also often termed *entity matching* or *entity resolution*. The goal is to identify entities from different KGs that refer to the same entity, e.g., Mount_Everest in DBpedia [40] and Q513 in Wikidata [79]. Conventional approaches to this task exploit a wide range of discriminative features of entities, e.g., names, descriptive annotations, and relational structures [15, 31, 32, 39, 70]. The major challenge lies in the symbolic, linguistic and schematic heterogeneity between independently-created KGs.

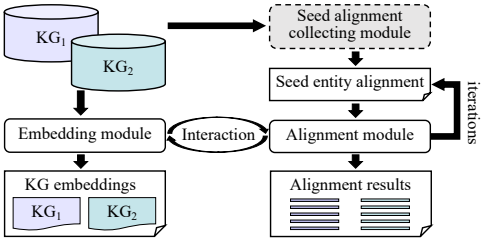


Figure 1: Framework of embedding-based entity alignment

Embedding-based entity alignment has emerged [10] and seen much development in recent years [8, 9, 24, 28, 57, 72, 73, 77, 81, 83, 93]. This approach is based on KG embedding techniques, which embed the symbolic representations of a KG as low-dimensional vectors in a way such that the semantic relatedness of entities is captured by the geometrical structures of an embedding space [5]. The premise is that such embeddings can potentially mitigate the aforementioned heterogeneity and simplify knowledge reasoning [80]. Figure 1 depicts a typical framework of embedding-based entity alignment. It takes as input two different KGs and collects *seed alignment* between them using sources such as the owl:sameAs links [10]. Then, the two KGs and seed alignment are fed into the *embedding* and *alignment* modules, to capture the correspondence of entity embeddings. There are two typical combination paradigms for module interaction: (i) the embedding module encodes the two KGs in *two* independent embedding spaces, meanwhile the alignment module uses seed alignment to learn a mapping between them [9, 10, 57, 58]; or (ii) the alignment module guides the embedding module to represent the two KGs into *one* unified space by forcing the aligned entities in seed alignment to hold very similar embeddings [8, 44, 72, 73, 77, 81, 93]. Finally, entity

*Wei Hu is the corresponding author.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 13, No. 11
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3407790.3407828

similarities are measured by the learned embeddings. We can predict the counterpart of a source entity through the nearest neighbor search among target entity embeddings using a distance metric like the Euclidean distance. Besides, to overcome the shortage of seed entity alignment, several approaches [9, 73, 93] deploy semi-supervised learning to iteratively augment new alignment.

However, as an emerging research topic, there are still some issues with analyzing and evaluating embedding-based entity alignment. First, as far as we know, there is no prior work summarizing the status quo of this field yet. The *latest development* of embedding-based entity alignment, as well as its *advantages* and *weaknesses* still remain to be explored. We even do not know how the embedding-based approaches compare to conventional entity alignment approaches. Second, there are also no widely-acknowledged benchmark datasets towards a realistic evaluation of embedding-based entity alignment. Arguably, a bit more popular datasets are DBP15K (used by [8, 42, 68, 72, 73, 81, 83, 84, 85, 87, 94])) and WK3L (used by [10, 44, 57, 58]). The different datasets for evaluation make it difficult to obtain a fair and comprehensive comparison of embedding-based entity alignment approaches. Moreover, current datasets contain much more high-degree entities (i.e., entities connected with many other entities, which are relatively easy for entity alignment) than real-world KGs do. As a result, many approaches may exhibit good performance on these biased datasets. Additionally, these datasets only focus on one aspect of heterogeneity, e.g., multilingualism, while overlook other aspects, e.g, different schemata and scales. This brings difficulties in understanding the generalization and robustness of embedding-based entity alignment. Third, we find that only a portion of the studies in this field come with source code, which makes it difficult to conduct further research on top of these approaches. Due to these issues, there is a pressing need to conduct a comprehensive and realistic re-evaluation of embedding-based entity alignment approaches with in-depth analysis.

In this paper, we carry out a systematic experimental study of embedding-based entity alignment with an open-source library. Our main contributions are listed as follows:

- *A comprehensive survey.* We survey 23 recent approaches for embedding-based entity alignment and categorize their core techniques and characteristics from different aspects. We also review the popular choices for each technical module, providing a brief overview of this field. (Sect. 2)

- *Benchmark datasets.* To make a fair and realistic comparison, we construct a set of dedicated benchmark datasets with splits of five folds by sampling real-world KGs DBpedia [40], Wikidata [79] and YAGO [61], in consideration of various aspects of heterogeneity regarding entity degrees, multilingualism, schemata and scales. Particularly, we propose a new sampling algorithm, which can make the properties (e.g., degree distribution) of a sample approximate its source KG. (Sect. 3)

- *Open-source library.* We develop an open-source library *OpenEA*[1] using Python and TensorFlow. This library integrates 12 representative embedding-based entity alignment approaches belonging to a wide range of technologies. It uses a flexible architecture to make it easy to integrate a large amount of existing KG embedding models (8 representative ones have been implemented) for entity alignment. The library will be duly updated along with the coming of new approaches, to facilitate future research. (Sect. 4)

- *Comprehensive comparison and analysis.* We provide a comprehensive comparison of 12 representative embedding-based entity alignment approaches in terms of both effectiveness and efficiency on our datasets. We train and tune each approach from scratch using our open-source library to ensure a fair evaluation. These results offer an overview of the performance of embedding-based entity alignment. To gain insights into the strengths and limitations of each approach, we conduct extensive analysis on their performance from different aspects. (Sect. 5)

- *Exploratory experiments.* We carry out three experiments beyond what has been available in literature. We give the first analysis on the geometric properties of entity embeddings to understand their underlying connections with the final performance. We notice that many KG embedding models have not been exploited for entity alignment and we explore 8 popular ones among them. We also compare embedding-based approaches with several conventional approaches, to explore their complementarity. (Sect. 6)

- *Future research directions.* Based on our survey and experimental findings, we provide a thorough outlook on several promising research directions for future work, including unsupervised entity alignment, long-tail entity alignment, large-scale entity alignment and entity alignment in non-Euclidean embedding spaces. (Sect. 7)

To the best of our knowledge, this work is the first systematic and comprehensive experimental study on embedding-based entity alignment between KGs. Our experiments reveal the true performance as well as the advantages and shortcomings of current approaches in the realistic entity alignment scenario. The shortcomings that we find, such as the incapacity of relation-based approaches in handling long-tail entities and the poor effectiveness of attribute-based approaches in resolving the heterogeneity of attribute values, call for the re-investigation of truly effective approaches for real-world entity alignment. We also believe that our in-depth analysis on the geometric properties of entity embeddings opens a new direction to investigate what enables the alignment-oriented embeddings and what supports the entity alignment performance behind the increasingly powerful approaches. Our benchmark datasets, library and experimental results are all publicly available through the GitHub repository[1] under the GPL license, to foster reproducible research. We think that the datasets and library will become a valuable and fundamental resource to future studies. As a growing number of knowledge-driven applications build their capacities on top of KGs and benefit from KG fusion, this work can lead to profound impacts to the KG and database communities.

2. PRELIMINARIES

We consider the entity alignment task between two KGs $\mathcal{KG}_1$ and $\mathcal{KG}_2$. Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote their entity sets, respectively. The goal is to find the 1-to-1 alignment of entities $\mathcal{S}_{\mathcal{KG}_1,\mathcal{KG}_2} = \{(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2 \,|\, e_1 \sim e_2\}$, where $\sim$ denotes an equivalence relation [39, 70]. In many cases, a small subset of the alignment $\mathcal{S}'_{\mathcal{KG}_1,\mathcal{KG}_2} \subset \mathcal{S}_{\mathcal{KG}_1,\mathcal{KG}_2}$, called *seed alignment*, is known beforehand and used as training data.

[1] https://github.com/nju-websoft/OpenEA

2.1 Literature Review

2.1.1 Knowledge Graph Embedding

Approaches. Existing KG embedding models can be generally divided into three categories: (i) *translational models*, e.g., TransE [5], TransH [82], TransR [49] and TransD [33]; (ii) *semantic matching models*, e.g., DistMult [86], ComplEx [76], HolE [54], SimplE [36], RotatE [71] and TuckER [3]; and (iii) *deep models*, e.g., ProjE [66], ConvE [13], R-GCN [63], KBGAN [7] and DSKG [25]. These models have been generally used for link prediction. We refer interested readers to the recent surveys [48, 80]. A related area is network embedding [26], which learns vertex representations to capture their proximity. However, the edges in networks carry simplex semantics. This differentiates network embedding from KG embedding in both data models and learning techniques.

Datasets & evaluation metrics. FB15K and WN18 are two benchmark datasets for link prediction in KGs [5]. Some studies notice that FB15K and WN18 suffer from the test leakage problem and build two new benchmark datasets FB15K-237 [75] and WN18RR [13] correspondingly. Three metrics are widely used in evaluation: (i) proportion of correct links in the top-m ranked results (called Hits@m, for example, $m = 1$), (ii) mean rank (MR) of correct links, and (iii) mean reciprocal rank (MRR). Two efforts in evaluating link prediction models have been reported in [1, 62].

2.1.2 Conventional Entity Alignment

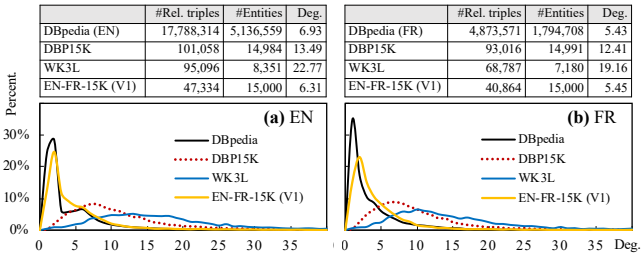
Approaches. Conventional approaches address entity alignment mainly from two angles. One is based on *equivalence reasoning* mandated by OWL semantics [22, 34]. The other is based on *similarity computation*, which compares symbolic features of entities [39, 65, 70]. Recent studies also use statistical machine learning [15, 31, 32] and crowdsourcing [96] to improve the accuracy. Also, in the database area, detecting duplicate entities, a.k.a. record linkage or entity resolution, has been extensively studied [16, 20]. These approaches mainly rely on literal information of entities.

Datasets & evaluation metrics. Since 2004, OAEI[2] (Ontology Alignment Evaluation Initiatives) has become the primary venue for work in ontology alignment. It also organizes an evaluation track for entity alignment in recent years. We have not observed any embedding-based systems participating in this track. The preferred evaluation metrics are precision, recall and F1-score.

2.1.3 Embedding-based Entity Alignment

Approaches. Many existing approaches [10, 47, 57, 58, 72, 73, 77, 93] employ the translational models (e.g., TransE [5]) to learn entity embeddings for alignment based on relation triples. Some recent approaches [8, 42, 81, 83, 85, 84, 88, 94] employ graph convolutional networks (GCNs) [38, 78]. Besides, some approaches incorporate attribute and value embeddings [9, 28, 72, 77, 83, 84, 87, 90]. We elaborate the techniques of these approaches in Sect. 2.2. Also, there are some approaches for (heterogeneous information) network alignment [29, 44, 89] or cross-lingual knowledge projection [56], which may also be modified for entity alignment. It is also worth noting that two studies [14, 52] design the embedding-based approaches for entity resolution in databases. They represent the attribute values of entities based on word embeddings and compare entities using embedding distances. However, they assume that all entities follow the same schema or the attribute alignment must be 1-to-1 mapping. As different KGs are often created with different schemata, it is hard to fulfill these requirements. Thus, they cannot be applied to entity alignment of KGs.

[2] http://oaei.ontologymatching.org/



	#Rel. triples	#Entities	Deg.
DBpedia (EN)	17,788,314	5,136,559	6.93
DBP15K	101,058	14,984	13.49
WK3L	95,096	8,351	22.77
EN-FR-15K (V1)	47,334	15,000	6.31

	#Rel. triples	#Entities	Deg.
DBpedia (FR)	4,873,571	1,794,708	5.43
DBP15K	93,016	14,991	12.41
WK3L	68,787	7,180	19.16
EN-FR-15K (V1)	40,864	15,000	5.45

Figure 2: Degree distributions and average degrees of two popular datasets DBP15K [72] and WK3L [10] used in previous approaches, along with our contributed dataset EN-FR-15K (V1). The x-axis denotes degrees and the y-axis denotes the percentage of entities w.r.t. degrees. These datasets are extracted from DBpedia [40], but the degree distributions of DBP15K and WK3L are quite different from DBpedia and their average degrees are also larger. Our dataset retains a similar degree distribution to DBpedia.

Datasets & evaluation metrics. To the best of our knowledge, there is no widely-acknowledged benchmark dataset for assessing embedding-based entity alignment approaches. Arguably, a bit more used datasets are DBP15K [72] and WK3L [10]. However, Figure 2 shows that their degree distributions and average degrees are significantly different from real-world KGs. More details about our datasets are reported in Sect. 3. Similar to link prediction, Hits@m, MR and MRR are mainly used as evaluation metrics, where Hits@1 should be emphasized, as it is equivalent to precision.

2.2 Categorization of Techniques

Table 1 categorizes 23 recent embedding-based entity alignment approaches by analyzing their embedding and alignment modules as well as the modes that they interact. For notations, we use capital calligraphic letters to denote sets and boldface letters for vectors and matrices.

2.2.1 Embedding Module

The embedding module seeks to encode a KG into a low-dimensional embedding space. Based on the types of triples used, we classify the KG embedding models in two types, i.e., *relation embedding* and *attribute embedding*. The former leverages relational learning techniques to capture KG structures, and the latter exploits attribute triples of entities.

Relation embedding is employed by all existing approaches. Below is three representative ways to realize it:

Triple-based embedding captures the local semantics of relation triples. Many KG embedding models fall into this category, which defines an energy function to measure the plausibility of triples. For example, TransE [5] interprets a relation as the translation from its head entity embedding to its tail. The energy of a relation triple (e_1, r_1, e_2) is

$$\phi(e_1, r_1, e_2) = \| \mathbf{e}_1 + \mathbf{r}_1 - \mathbf{e}_2 \|, \quad (1)$$

where $\| \cdot \|$ denotes the L_1- or L_2-norm of vectors. TransE optimizes the *marginal* ranking loss to separate positive triples from negatives by a pre-defined margin. Other choices

Table 1: Categorization of popular embedding-based entity alignment approaches published before December 2019

	Embedding		Alignment	Interaction	
	Relation	Att.	Emb. distance	Combination	Learning
MTransE [10]	Triple	-	Euclidean	Transformation	Superv.
IPTransE [93]	Path	-	Euclidean	Sharing	Semi-
JAPE [72]	Triple	Att.	Cosine	Sharing	Superv.
BootEA [73]	Triple	-	Cosine	Swapping	Semi-
KDCoE [9]	Triple	Literal	Euclidean	Transformation	Semi-
NTAM [44]	Triple	-	Cosine	Swapping	Superv.
GCNAlign [81]	Neighbor	Att.	Manhattan	Calibration	Superv.
AttrE [77]	Triple	Literal	Cosine	Sharing	Superv.
IMUSE [28]	Triple	Literal	Cosine	Sharing	Superv.
SEA [57]	Triple	-	Cosine	Transformation	Superv.
RSN4EA [24]	Path	-	Cosine	Sharing	Superv.
GMNN [85]	Neighbor	Literal	Cosine	Swapping	Superv.
MuGNN [8]	Neighbor	-	Manhattan	Calibration	Superv.
OTEA [58]	Triple	-	Euclidean	Transformation	Superv.
NAEA [94]	Neighbor	-	Cosine	Swapping	Superv.
AVR-GCN [88]	Neighbor	-	Euclidean	Swapping	Superv.
MultiKE [90]	Triple	Literal	Cosine	Swapping	Superv.
RDGCN [83]	Neighbor	Literal	Manhattan	Calibration	Superv.
KECG [42]	Neighbor	-	Euclidean	Calibration	Superv.
HGCN [84]	Neighbor	Literal	Euclidean	Calibration	Superv.
MMEA [68]	Triple	-	Cosine	Sharing	Superv.
HMAN [87]	Neighbor	Literal	Euclidean	Calibration	Superv.
AKE [47]	Triple	-	Euclidean	Transformation	Superv.

of loss functions include the *logistic* loss [54, 76] and the *limit-based* loss [73, 91]. Negative triples can be generated using the *uniform* negative sampling or *truncated* sampling.

Path-based embedding exploits the long-term dependency of relations spanning over relation paths. A relation path is a set of nose-to-tail linked relation triples, e.g., $(e_1, r_1, e_2), (e_2, r_2, e_3)$. IPTransE [93] models relation paths by inferring the equivalence between a direct relation and a multi-hop path. Assume that there is a direct relation r_3 from e_1 to e_3. IPTransE expects the embedding of r_3 to be similar to the path embedding, which is encoded as a combination of its constituent relation embeddings:

$$\mathbf{r}^* = \text{comb}(\mathbf{r}_1, \mathbf{r}_2), \tag{2}$$

where $\text{comb}(\cdot)$ is a sequence composition operation such as sum. $\| \mathbf{r}^* - \mathbf{r}_3 \|$ is minimized to make them close to each other. However, IPTransE overlooks entities. Another work, RSN4EA [24], modifies recurrent neural networks (RNNs) to model the sequence of entities and relations together.

Neighborhood-based embedding uses the subgraph structure constituted by a large amount of relations between entities. GCNs [6, 12, 38, 63] are well suited for modeling this structure, and have been used for embedding-based entity alignment recently [8, 42, 81, 83, 84, 85, 87]. A GCN consists of multiple graph convolutional layers. Let $\mathbf{A}$ denote the adjacency matrix of a KG and $\mathbf{H}^{(0)}$ be a feature matrix where each row corresponds to an entity. The typical propagation rule from the i^{th} layer to the $(i+1)^{\text{th}}$ layer [38] is

$$\mathbf{H}^{(i+1)} = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(i)}\mathbf{W}), \tag{3}$$

with $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\mathbf{I}$ is an identity matrix. $\hat{\mathbf{D}}$ is the diagonal degree matrix of $\hat{\mathbf{A}}$. $\mathbf{W}$ is the learnable weight matrix. $\sigma(\cdot)$ is the activation function such as $\tanh(\cdot)$.

Attribute embedding is used by several approaches [9, 28, 72, 77, 81, 83, 85, 87, 90] to enhance the similarity measure of entities. There are two ways for attribute embedding:

Attribute correlation embedding considers the correlations among attributes. Attributes are regarded as correlated if they are frequently used together to describe an entity. For example, longitude is highly correlated with latitude as they often form a coordinate. JAPE [72] exploits such correlations for entity alignment, based on the assumption that similar entities should have similar correlated attributes. For two attributes a_1, a_2, the probability that they are correlated is

$$\Pr(a_1, a_2) = \text{sigmoid}(\mathbf{a}_1 \cdot \mathbf{a}_2), \tag{4}$$

where attribute embeddings can be learned by maximizing the probability over all attribute pairs. Here, the attribute correlation embedding does not consider literal values.

Literal embedding introduces literal values to attribute embedding. AttrE [77] proposes a character-level encoder that is capable of dealing with unseen values in training phases. Let $v = (c_1, c_2, ..., c_n)$ be a literal with n characters, where c_i $(1 \le i \le n)$ is the i^{th} character. AttrE embeds v as

$$\mathbf{v} = \text{comb}(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n). \tag{5}$$

With this representation, literals are treated as entities and the relation embedding models like TransE can be used to learn from attribute triples. However, the character-based literal embedding may fail in cross-lingual settings.

2.2.2 Alignment Module

The alignment module uses seed alignment as labeled training data to capture the correspondence of entity embeddings. Two keys are picking a distance metric and designing an alignment inference strategy.

Distance metrics. *Cosine, Euclidean and Manhattan* distances are three widely-used metrics. In high-dimensional spaces, a few vectors (called hubs [60]) may repeatedly occur as the k-nearest neighbors of others, the so-called *hubness* problem [11]. See Sect. 6.1 for more details.

Alignment inference strategies. *Greedy search* is used by all current approaches. Given $\mathcal{KG}_1$ and $\mathcal{KG}_2$ to be aligned and a distance metric π, for each entity $e_1 \in \mathcal{E}_1$, it finds the aligned entity $\tilde{e_2}$ by $\tilde{e_2} = \arg\min_{e_2 \in \mathcal{E}_2} \pi(\mathbf{e}_1, \mathbf{e}_2)$. Differently, *collective search* [37, 51] aims to find a global optimal alignment that minimizes $\sum_{(e_1,e_2)\in\mathcal{S}_{\mathcal{KG}_1,\mathcal{KG}_2}} \pi(\mathbf{e}_1, \mathbf{e}_2)$. It can be modeled as the maximum weight matching problem in a bipartite graph and solved in $O(N^3)$ time using the Kuhn-Munkres algorithm ($N = |\mathcal{E}_1| + |\mathcal{E}_2|$), or reduced to linear time using the heuristic algorithm [30]. Another solution is the stable marriage algorithm [50]. The alignment between $\mathcal{E}_1$ and $\mathcal{E}_2$ satisfies a stable marriage if there does not exist a pair of entities that both prefer each other than their current aligned ones. Its solution takes $O(N^2)$ time [17].

2.2.3 Interaction Mode

Combination modes. Four typical designs to reconcile KG embeddings for entity alignment are as follows: *Embedding space transformation* embeds two KGs in different embedding spaces and learns a transformation matrix $\mathbf{M}$ between the two spaces using seed alignment, to achieve $\mathbf{M}\mathbf{e}_1 \approx \mathbf{e}_2$ for each $(e_1, e_2) \in \mathcal{S}'_{\mathcal{KG}_1,\mathcal{KG}_2}$. Another combination mode encodes two KGs into a unified embedding space. *Embedding space calibration* minimizes $\| \mathbf{e}_1 - \mathbf{e}_2 \|$ for each $(e_1, e_2) \in \mathcal{S}'_{\mathcal{KG}_1,\mathcal{KG}_2}$ to calibrate the embeddings of seed alignment. As two special cases, *parameter sharing* directly configures $\mathbf{e}_1 = \mathbf{e}_2$ and *parameter swapping* swaps seed entities in their triples to generate extra triples as supervision. For instance, given $(e_1, e_2) \in \mathcal{S}'_{\mathcal{KG}_1,\mathcal{KG}_2}$ and a relation triple of $\mathcal{KG}_1$ (e_1, r_1, e'_1), parameter swapping produces a new triple (e_2, r_1, e'_1) and feeds it in KG embedding models as a real triple. Both parameter sharing and swapping methods do not introduce new loss functions, but the latter produces more triples.

Algorithm 1: Iterative degree-based sampling (IDS)

Input: $\mathcal{KG}_1, \mathcal{KG}_2$, reference alignment $\mathcal{S}_{ref}$, entity size N, hyper-parameters μ, ϵ

```
// only retain entities in reference alignment
```
1 Filter $\mathcal{KG}_1, \mathcal{KG}_2$ by $\mathcal{S}_{ref}$;
2 Get degree distributions Q_1, Q_2 for $\mathcal{KG}_1, \mathcal{KG}_2$, resp.;
3 **do** `// if fails, run it again`
4 Initialize datasets $\mathcal{DS}_1, \mathcal{DS}_2$ from $\mathcal{KG}_1, \mathcal{KG}_2$, resp.;
5 **while** $|\mathcal{DS}_1| > N$ && $|\mathcal{DS}_2| > N$ **do**
6 **for** $\mathcal{DS}_j$ $(j = 1, 2)$ **do**
7 Get $dsize_j(x, \mu)$ for each degree x;
8 Get entity deletion probability by PageRank;
9 Delete $dsize_j(x, \mu)$ entities w.r.t. probabilities;
10 Filter $\mathcal{DS}_1, \mathcal{DS}_2$ by $\mathcal{S}_{ref}$; update $\mathcal{S}_{ref}$ accordingly;
11 Get degree distributions P_1, P_2 for $\mathcal{DS}_1, \mathcal{DS}_2$, resp.;
12 **while** $JS(Q_1, P_1) > \epsilon \parallel JS(Q_2, P_2) > \epsilon$;
13 **return** $\mathcal{DS}_1, \mathcal{DS}_2, \mathcal{S}_{ref}$;

Learning strategies. Based on how to process labeled and unlabeled data, learning strategies can be divided below:

Supervised learning leverages the seed alignment as labeled training data. For embedding space transformation, seed alignment is used to learn the transformation matrix. For space calibration, it is used to let aligned entities have similar embeddings. But, the acquisition of seed alignment is costly and error-prone, especially for cross-lingual KGs.

Semi-supervised learning uses unlabeled data in training, e.g., self-training [73, 93] and co-training [9]. The former iteratively proposes new alignment to augment seed alignment. The latter combines two models learned from disjoint entity features and alternately enhances the alignment learning of each other. Although OTEA [58] and KECG [42] claim that they are semi-supervised approaches, their learning strategies do not augment seed alignment. We do not treat them as standard semi-supervised learning in this paper.

Unsupervised learning needs no training data. We have not observed any embedding-based entity alignment approaches using unsupervised learning. Although IMUSE [28] claims that it is an unsupervised approach, it actually uses a preprocessing method to collect seed alignment with high string similarity. Its embedding module still needs seed alignment.

3. DATASET GENERATION

As aforementioned, current widely-used datasets are quite different from real-world KGs. Also, it is hard for embedding-based approaches to run on full KGs due to the large and unpartitioned candidate space. Hence, we sample real-world KGs and provide two data scales (15K and 100K).

3.1 Iterative Degree-based Sampling

We consider five factors in building our datasets: source KGs, reference alignment, dataset sizes, languages and density, where the last is more challenging for building datasets. Specifically, we want to generate a certain-sized dataset from a source KG such that the difference of their entity degree distributions does not exceed an expectation. The difficulty lies in that the removal of an entity from the source KG also changes the connectivity of its neighboring entities.

We propose an iterative degree-based sampling (IDS) algorithm, which simultaneously deletes entities in two source KGs with reference alignment until achieving the desired size, meanwhile keeping a similar degree distribution of each sampled dataset as the source KG. Algorithm 1 describes the sampling procedure. During iterations, the proportion of entities having degree x in the current dataset, denoted by $P(x)$, cannot always equal the original proportion $Q(x)$. We adjust the entity size to be deleted by $dsize(x, \mu) = \mu\big(1 + P(x) - Q(x)\big)$, where μ is the base step size (see Line 7). Moreover, we prefer not to delete entities having a big influence on the overall degree distribution, such as the ones of high degree. To achieve this, we leverage the PageRank value for measuring the probability of an entity to be deleted (Line 8).

We use the Jensen-Shannon (JS) divergence [46] to assess the difference of two degree distributions (Line 12). Given two degree distributions Q, P, their JS-divergence is:

$$JS(Q, P) = \frac{1}{2} \sum_{x=1}^{n} \Big(Q(x) \log \frac{Q(x)}{M(x)} + P(x) \log \frac{P(x)}{M(x)} \Big), \quad (6)$$

where $Q(x)$ and $P(x)$ denote the proportions of entities with degree x $(x = 1 \dots n)$ in Q, P, respectively, and $M = \frac{Q+P}{2}$. A small JS divergence ϵ between Q and P reveals that they have similar degree distributions. We set expectation $\epsilon \leq 5\%$. The most costly part of IDS is to calculate PageRank weights during the iterations of deleting entities. It can be scaled to very large KGs by using approximation algorithms [2].

3.2 Dataset Overview

We choose three well-known KGs as our sources: DBpedia (2016-10) [40], Wikidata (20160801) [79] and YAGO 3 [61]. Also, we consider two cross-lingual versions of DBpedia: English–French and English–German. We follow the conventions in [10, 72, 73, 81, 93] to generate datasets of two sizes with 15K and 100K entities, using the IDS algorithm. Specifically, we make use of DBpedia's inter-language links and owl:sameAs among the three KGs to retrieve reference entity alignment. To balance the efficiency and deletion safety, we set $\mu = 100$ for 15K and $\mu = 500$ for 100K.

The statistics of the datasets are listed in Table 2. We generate two versions of datasets for each pair of source KGs. V1 is gained by directly using the IDS algorithm. For V2, we first randomly delete entities with low degrees ($d \leq 5$) in the source KG to make the average degree doubled, and then execute IDS to fit the new KG. As a result, V2 is twice denser than V1 and more similar to existing datasets [10, 72]. Figure 3 shows the degree distributions and average degrees of EN-FR-15K (V1, V2) and EN-FR-100K (V1, V2). Our 15K and 100K datasets are much closer to the source KGs.

For each dataset, we also extract the attribute triples of entities to fulfill the input requirement of some approaches [9, 28, 72, 77, 81, 83, 85, 90]. Considering that DBpedia, Wikidata and YAGO collect data from very similar sources (mainly, Wikipedia), the aligned entities usually have identical labels. They would become "tricky" features for entity alignment and influence the evaluation of real performance. According to the suggestion in [95], we delete entity labels.

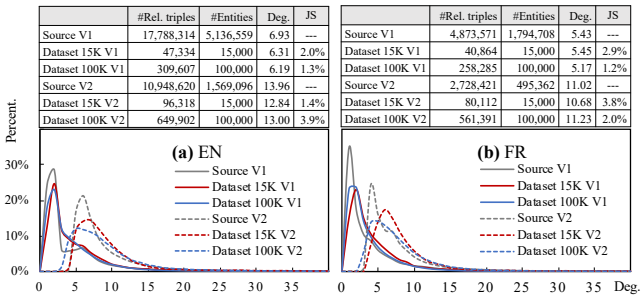
By convention, we split a dataset into training, validation and test sets. The details are given in Sect. 5.1.

3.3 Dataset Evaluation

We assess IDS and the quality of our datasets. Note that, generating an entity alignment dataset is a non-trivial work, as a qualified dataset needs to hold several characteristics, such as good connectivity (due to many approaches rely on graph structures), similar degree distributions to original KGs (for a realistic entity alignment scenario), and enough alignment (for training/validation/test). As far as we know, there still lacks a sampling method dedicated to this problem.

Table 2: Dataset statistics

Datasets	KGs	15K (V1)				15K (V2)				100K (V1)				100K (V2)			
		#Rel.	#Att.	#Rel tr.	#Att tr.	#Rel.	#Att.	#Rel tr.	#Att tr.	#Rel.	#Att.	#Rel tr.	#Att tr.	#Rel.	#Att.	#Rel tr.	#Att tr.
EN-FR	EN	267	308	47,334	73,121	193	189	96,318	66,899	400	466	309,607	497,729	379	364	649,902	503,922
	FR	210	404	40,864	67,167	166	221	80,112	68,779	300	519	258,285	426,672	287	468	561,391	431,379
EN-DE	EN	215	286	47,676	83,755	169	171	84,867	81,988	381	451	335,359	552,750	323	326	622,588	560,247
	DE	131	194	50,419	156,150	96	116	92,632	186,335	196	252	336,240	716,615	170	189	629,395	793,710
D-W	DB	248	342	38,265	68,258	167	175	73,983	66,813	413	493	293,990	451,011	318	328	616,457	467,103
	WD	169	649	42,746	138,246	121	457	83,365	175,686	261	874	251,708	687,860	239	760	588,203	878,219
D-Y	DB	165	257	30,291	71,716	72	90	68,063	65,100	287	379	294,188	523,062	230	277	576,547	547,026
	YG	28	35	26,638	132,114	21	20	60,970	131,151	32	38	400,518	749,787	31	36	865,265	855,161



	#Rel. triples	#Entities	Deg.	JS
Source V1	17,788,314	5,136,559	6.93	---
Dataset 15K V1	47,334	15,000	6.31	2.0%
Dataset 100K V1	309,607	100,000	6.19	1.3%
Source V2	10,948,620	1,569,096	13.96	---
Dataset 15K V2	96,318	15,000	12.84	1.4%
Dataset 100K V2	649,902	100,000	13.00	3.9%

	#Rel. triples	#Entities	Deg.	JS
Source V1	4,873,571	1,794,708	5.43	---
Dataset 15K V1	40,864	15,000	5.45	2.9%
Dataset 100K V1	258,285	100,000	5.17	1.2%
Source V2	2,728,421	495,362	11.02	---
Dataset 15K V2	80,112	15,000	10.68	3.8%
Dataset 100K V2	561,391	100,000	11.23	2.0%

Figure 3: Degree distributions and average degrees of our sampled datasets EN-FR-15K (V1, V2) and EN-FR-100K (V1, V2), compared with DBpedia (the source KG)

Table 3: Comparison of the EN-FR-15K (V1) datasets generated by RAS, PRS and IDS

Datasets	KGs	#Alignment	Deg.	JS	Isolates	Cluster coef.
DBpedia	EN	525,807	6.39	–	0	0.342
	FR		5.43	–	0	0.080
RAS	EN	15,000	0.27	14.5%	85.5%	0.002
	FR		0.17	12.1%	90.1%	0.001
PRS	EN	15,000	1.20	7.3%	68.9%	0.025
	FR		0.63	9.3%	69.4%	0.015
IDS	EN	15,000	6.31	2.0%	0	0.233
	FR		5.45	2.9%	0	0.190

For evaluation, we design two baseline methods on the basis of existing graph sampling algorithms [41]:

- *Random alignment sampling (RAS)* first randomly selects a fixed size (e.g., 15K) of entity alignment between two KGs, and then extracts the relation triples whose head and tail entities are both in the sampled entities.
- *PageRank-based sampling (PRS)* first samples entities from one KG based on the PageRank scores (entities not involved in any alignment are discarded), and then extracts these entities' counterparts from the other KG.

Table 3 lists the properties of EN-FR-15K (V1) datasets generated by RAS, PRS and our IDS, compared to the source KGs (relation triples). In addition to the average degree and JS-divergence, we further consider two metrics: percentage of isolated entities [19] and clustering coefficient [41]. The dataset of RAS is much sparser than the source, because the random sampling cannot retain the connectivity and degree distribution [69]. It has a low clustering coefficient and contains many isolated entities that are typically hard for embedding modules to handle. PRS more focuses on high-degree entities and gets better properties than RAS. However, the dataset is still far away from satisfactory due to the low average degree, high JS value and high percentage of isolated entities. This is because its entity selection procedure only applies to one KG rather than two KGs together. Differently, IDS considers the degree distributions of two KGs together. It tends to sample two aligned entities with similar degrees. Thus, the two KGs of our dataset have similar clustering coefficients. As the sampled dataset is much smaller that the source, it is hard to keep all these properties well. IDS shows good comprehensive performance.

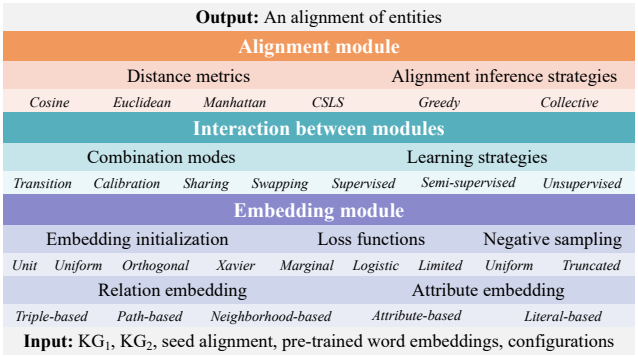


Figure 4: Software architecture of OpenEA

4. OPEN-SOURCE LIBRARY

We use Python and TensorFlow to develop an open-source library, namely **OpenEA**, for embedding-based entity alignment. The software architecture is illustrated in Figure 4. Our design goals and features include three aspects:

Loose coupling. The implementation of embedding and alignment modules is independent to each other. OpenEA provides a framework template with pre-defined input and output data structures to make these modules as an integral pipeline. Users can freely call and combine different techniques in these modules to develop new approaches.

Functionality and extensibility. OpenEA implements a set of necessary functions as its underlying components, including initialization functions, loss functions and negative sampling methods in the embedding module; combination and learning strategies in the interaction mode; as well as distance metrics and alignment inference strategies in the alignment module. On top of those, OpenEA also provides a set of flexible and high-level functions with configuration options to call these components. In this way, new functions can be easily integrated by adding new configuration options.

Off-the-shelf approaches. To facilitate the usage of OpenEA and support our experimental study, we try our best to integrate or rebuild 12 representative embedding-based entity alignment approaches belonging to a wide range of technologies, including MTransE, IPTransE, JAPE, KDCoE, BootEA, GCNAlign, AttrE, IMUSE, SEA, RSN4EA, MultiKE and RDGCN. MTransE, JAPE, KDCoE, BootEA, GCNAlign, AttrE, RSN4EA, MultiKE and RDGCN are implemented by integrating their source code, while IPTransE, IMUSE and SEA are rebuilt by ourselves. Moreover, we integrate several relation embedding models that have not been explored for entity alignment yet, including three translational models TransH [82], TransR [49] and TransD [33]; three semantic matching models HolE [54], SimplE [36] and RotatE [71];

Table 4: Common hyper-parameters for all the approaches

	15K	100K
Batch size for rel. trip.	5,000	20,000
Termination condition	Early stop when the Hits@1 score begins to drop on the validation sets, checked every 10 epochs.	
Max. epochs	2000	

as well as two deep models ProjE [66] and ConvE [13]. We also integrate two attribute embedding models AC2Vec [72] and Label2Vec [90], based on pre-trained multilingual word embeddings [4]. TransH, TransR, TransD and HolE are developed by referring to the open-source toolkit OpenKE [27]; the remaining is implemented based on their source code.

5. EXPERIMENTS AND RESULTS

In this section, we report a comprehensive evaluation using our benchmark datasets and open-source library.

5.1 Experiment Settings

Environment. We carry out the experiments on the workstation with an Intel Xeon E3 3.3GHz CPU, 128GB memory, a NVIDIA GeForce GTX 1080Ti GPU and Ubuntu 16.04.

Cross-validation. We conduct the experiments with 5-fold cross-validation to ensure unbiased evaluation. Specifically, we divide the reference entity alignment into five disjoint folds, each of which accounts for 20% of the total. For each running, we pick one fold (20%) as training data and leave the remaining for validation (10%) and testing (70%). As found in [10], the inter-language links in the multilingual Wikipedia cover about 15% of entity alignment. Thus, using 20% as training data can both satisfy the need for 5-fold cross-validation and conform to the real world.

Comparative approaches and settings. We evaluate all the embedding-based entity alignment approaches implemented in OpenEA. To make a fair comparison, we use our best efforts to unify the experiment settings. Table 4 shows the common hyper-parameters used for all the approaches. As indicated in [35], the batch size has an influence on the performance and running time. So, we use a fixed batch size for relation triples to avoid its interference. For other settings specific to each approach, we follow the reported details in literature as carefully as we can, e.g., the margin for the ranking loss in IPTransE and AttrE is 1.5; the number of GCN layers in GCNAlign and RDGCN is 2. For several key hyper-parameters and the unreported ones, we try our best to tune them. For example, we constrain the L_2-norm of entity embeddings to 1 for many approaches, e.g., IMUSE, because we find that such normalization yields better results. For cross-lingual datasets, we use pre-trained cross-lingual word embeddings [4] to initialize literal embeddings for the approaches using attribute values. The hyper-parameter settings of each approach on our datasets are available online. Notice that there are emerging approaches (e.g., AliNet [74]) that are contemporaneous to this paper. We will accordingly include those approaches into future release of OpenEA.

Evaluation metrics. In our experiments, the default alignment direction is from left to right. Take D-W for example. We treat DBpedia as the source and align it with the target KG Wikidata. Following the conventions, we use Hits@m ($m = 1, 5$), MR and MRR as the evaluation metrics.

Availability. We release the datasets and OpenEA library online. The experimental results on five folds of each dataset using all the metrics are provided in the CSV format. All will be duly updated along with the coming of new approaches.

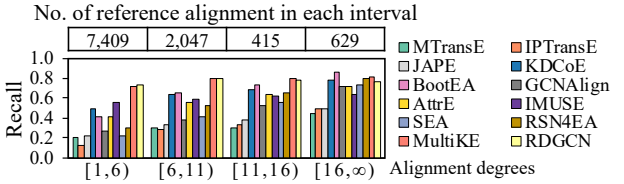


Figure 5: Recall w.r.t. alignment deg. on EN-FR-15K (V1)

5.2 Main Results and Analysis

Table 5 depicts the Hits@1, Hits@5 and MRR results of the 12 implemented approaches on our datasets. In summary, RDGCN, BootEA and MultiKE achieve the top-3 results. For a comprehensive and thorough understanding, we analyze the results from five angles:

Sparse datasets (V1) vs. dense datasets (V2). From Table 5, we find that most relation-based approaches perform better on the dense datasets than on the sparse ones, e.g., IPTransE, BootEA, SEA and RSN4EA. This is in accord with our intuition that the entities in the dense datasets are generally involved in more relation triples, which enable these approaches to capture more semantic information. For the approaches considering attribute triples, KDCoE, GCNAlign, AttrE, IMUSE and RDGCN also perform better on the dense datasets, indicating that the relation embeddings still make contributions. Differently, MultiKE relies on multiple "views" of features, which make it relatively insensitive to the relation changes. Interestingly, we also see that the performance of two relation-based approaches, MTransE and JAPE, drops on some dense datasets. We believe that this is because they are based on TransE, which has deficiency in handling multi-mapping relations in the dense datasets. For example, 39.0% of entities in EN-FR-100K (V1) have multi-mapping relations while the proportion in EN-FR-100K (V2) reaches up to 71.2%. The complex structures make MTransE and JAPE prone to learn very similar embeddings for different entities involving the same multi-mapping relation [49, 82].

For further analysis, we divide the test alignment of each dataset into multiple groups in terms of alignment degrees. The degree of an alignment is defined as the sum of relation triples for the two involved entities. Figure 5 illustrates the recall results on EN-FR-15K (V1). Obviously, most entities have relatively few relation triples, and we call them *long-tail* entities. *We find that all the relation-based approaches run better in aligning entities with rich relation triples while their results decline on long-tail entities*, as long-tail entities have little information useful for learning, which limits the expressiveness of their embeddings. This lopsided performance confirms the results on the sparse and dense datasets from another angle. By using additional literals, the lopsided performance of KDCoE, AttrE, IMUSE, MultiKE and RDGCN alleviates. However, JAPE and GCNAlign that use attribute correlations still show the lopsided performance for entities with different degrees. The experiments on other datasets also agree on the above observations. Currently, we have not seen a method that handles long-tail entities well.

15K datasets vs. 100K datasets. We observe that all the approaches perform better on the 15K datasets than on the 100K datasets, except D-Y, because the 100K datasets have more complex structures, causing more difficulties for embedding-based approaches to capture entity proximity. For example, 34.9% of entities in EN-FR-15K (V1) are involved in multi-mapping relations while the proportion in EN-FR-100K (V1) reaches 39.0%. As we have discussed, multi-mapping

Table 5: Cross-validation results of current representative approaches on the 15K and 100K datasets

		15K (V1)			15K (V2)			100K (V1)			100K (V2)		
		Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR	Hits@1	Hits@5	MRR
EN-FR	MTransE	.247 ± .006	.467 ± .009	.351 ± .007	.240 ± .005	.436 ± .007	.336 ± .005	.138 ± .002	.261 ± .004	.202 ± .002	.090 ± .003	.174 ± .003	.135 ± .003
	IPTransE	.169 ± .013	.320 ± .025	.243 ± .019	.236 ± .012	.449 ± .021	.339 ± .016	.158 ± .004	.277 ± .008	.219 ± .006	.234 ± .007	.431 ± .015	.329 ± .010
	JAPE	.262 ± .006	.497 ± .010	.372 ± .007	.292 ± .009	.524 ± .006	.402 ± .007	.165 ± .002	.310 ± .002	.240 ± .002	.125 ± .003	.239 ± .005	.183 ± .004
	KDCoE	.581 ± .004	.680 ± .004	.628 ± .003	.730 ± .007	.837 ± .006	.778 ± .005	.482 ± .005	.515 ± .006	.499 ± .005	.611 ± .012	.653 ± .015	.632 ± .014
	BootEA	.507 ± .010	.718 ± .012	.603 ± .011	.660 ± .006	.850 ± .005	.745 ± .005	.389 ± .004	.561 ± .004	.474 ± .004	.640 ± .001	.806 ± .001	.716 ± .000
	GCNAlign	.338 ± .002	.589 ± .009	.451 ± .005	.414 ± .005	.698 ± .007	.542 ± .005	.230 ± .002	.412 ± .004	.319 ± .003	.257 ± .002	.455 ± .003	.351 ± .002
	AttrE	.481 ± .010	.671 ± .009	.569 ± .010	.535 ± .015	.746 ± .014	.631 ± .014	.403 ± .019	.572 ± .019	.483 ± .019	.466 ± .011	.644 ± .012	.549 ± .011
	IMUSE	.569 ± .006	.717 ± .010	.638 ± .008	.607 ± .013	.760 ± .014	.678 ± .013	.439 ± .002	.546 ± .004	.492 ± .003	.461 ± .003	.605 ± .005	.529 ± .004
	SEA	.280 ± .015	.530 ± .026	.397 ± .019	.360 ± .018	.651 ± .018	.494 ± .017	.225 ± .011	.399 ± .013	.314 ± .012	.297 ± .002	.500 ± .002	.395 ± .002
	RSN4EA	.393 ± .007	.595 ± .012	.487 ± .009	.579 ± .006	.759 ± .006	.662 ± .006	.293 ± .004	.452 ± .006	.371 ± .004	.495 ± .003	.672 ± .005	.578 ± .004
	MultiKE	.749 ± .004	.819 ± .005	.782 ± .004	.864 ± .007	.909 ± .005	.885 ± .006	.629 ± .002	.680 ± .002	.655 ± .002	.642 ± .003	.696 ± .003	.670 ± .003
	RDGCN	.755 ± .004	.854 ± .003	.800 ± .003	.847 ± .006	.919 ± .004	.880 ± .005	.640 ± .004	.732 ± .004	.683 ± .004	.715 ± .003	.787 ± .002	.748 ± .002
EN-DE	MTransE	.307 ± .007	.518 ± .004	.407 ± .006	.193 ± .016	.352 ± .023	.274 ± .018	.140 ± .003	.264 ± .004	.204 ± .004	.115 ± .003	.215 ± .004	.168 ± .003
	IPTransE	.350 ± .009	.515 ± .012	.43 ± .011	.476 ± .012	.678 ± .011	.571 ± .010	.226 ± .014	.357 ± .019	.292 ± .017	.346 ± .013	.535 ± .016	.437 ± .014
	JAPE	.288 ± .016	.512 ± .018	.394 ± .016	.167 ± .011	.329 ± .015	.250 ± .013	.152 ± .006	.291 ± .009	.223 ± .007	.11 ± .004	.218 ± .006	.167 ± .005
	KDCoE	.529 ± .014	.629 ± .015	.580 ± .014	.649 ± .017	.788 ± .017	.715 ± .016	.506 ± .014	.591 ± .019	.549 ± .016	.651 ± .011	.756 ± .010	.701 ± .011
	BootEA	.675 ± .004	.820 ± .004	.740 ± .004	.833 ± .015	.912 ± .008	.869 ± .012	.518 ± .003	.673 ± .003	.592 ± .003	.739 ± .004	.851 ± .003	.791 ± .004
	GCNAlign	.481 ± .003	.679 ± .005	.571 ± .003	.534 ± .005	.717 ± .005	.618 ± .005	.317 ± .007	.485 ± .008	.399 ± .007	.375 ± .005	.549 ± .006	.457 ± .005
	AttrE	.517 ± .011	.687 ± .013	.597 ± .011	.650 ± .015	.816 ± .008	.726 ± .012	.399 ± .010	.554 ± .012	.473 ± .011	.464 ± .011	.637 ± .010	.546 ± .011
	IMUSE	.580 ± .017	.720 ± .014	.647 ± .015	.674 ± .011	.803 ± .008	.734 ± .010	.421 ± .005	.516 ± .005	.469 ± .005	.457 ± .005	.588 ± .007	.521 ± .006
	SEA	.530 ± .027	.718 ± .026	.617 ± .025	.606 ± .024	.779 ± .018	.687 ± .020	.341 ± .016	.502 ± .017	.421 ± .016	.447 ± .006	.625 ± .006	.532 ± .006
	RSN4EA	.587 ± .001	.752 ± .003	.662 ± .001	.791 ± .009	.890 ± .006	.837 ± .008	.430 ± .002	.57 ± .001	.497 ± .001	.639 ± .001	.763 ± .001	.697 ± .001
	MultiKE	.756 ± .004	.809 ± .003	.782 ± .003	.755 ± .008	.813 ± .008	.784 ± .007	.668 ± .002	.712 ± .002	.690 ± .001	.661 ± .004	.709 ± .004	.686 ± .004
	RDGCN	.830 ± .006	.895 ± .004	.859 ± .005	.833 ± .007	.891 ± .005	.860 ± .006	.722 ± .002	.794 ± .002	.756 ± .002	.766 ± .002	.829 ± .002	.796 ± .002
D-W	MTransE	.259 ± .008	.461 ± .012	.354 ± .008	.271 ± .013	.49 ± .014	.376 ± .013	.210 ± .003	.358 ± .003	.282 ± .003	.148 ± .004	.268 ± .005	.209 ± .005
	IPTransE	.232 ± .012	.38 ± .016	.303 ± .014	.412 ± .007	.623 ± .010	.511 ± .007	.221 ± .004	.352 ± .008	.285 ± .006	.319 ± .017	.516 ± .024	.413 ± .020
	JAPE	.250 ± .007	.457 ± .010	.348 ± .007	.262 ± .013	.484 ± .019	.368 ± .015	.211 ± .004	.369 ± .004	.287 ± .004	.154 ± .004	.287 ± .005	.221 ± .005
	KDCoE	.247 ± .020	.412 ± .029	.325 ± .023	.405 ± .020	.640 ± .019	.515 ± .020	.157 ± .003	.243 ± .007	.199 ± .005	.373 ± .010	.550 ± .014	.458 ± .012
	BootEA	.572 ± .008	.744 ± .007	.649 ± .008	.821 ± .004	.926 ± .003	.867 ± .003	.516 ± .006	.685 ± .006	.594 ± .005	.766 ± .007	.892 ± .005	.822 ± .006
	GCNAlign	.364 ± .009	.580 ± .010	.461 ± .008	.506 ± .006	.743 ± .005	.612 ± .005	.324 ± .002	.507 ± .004	.409 ± .003	.353 ± .004	.559 ± .006	.449 ± .004
	AttrE	.299 ± .004	.467 ± .003	.381 ± .003	.489 ± .016	.695 ± .016	.585 ± .015	.209 ± .008	.335 ± .011	.273 ± .009	.301 ± .015	.475 ± .018	.386 ± .016
	IMUSE	.327 ± .016	.523 ± .024	.419 ± .019	.581 ± .016	.778 ± .011	.671 ± .014	.276 ± .010	.437 ± .016	.355 ± .013	.431 ± .011	.631 ± .013	.525 ± .012
	SEA	.360 ± .012	.572 ± .015	.458 ± .013	.567 ± .008	.770 ± .007	.660 ± .008	.291 ± .012	.470 ± .014	.378 ± .013	.382 ± .003	.585 ± .003	.479 ± .002
	RSN4EA	.441 ± .008	.615 ± .007	.521 ± .007	.723 ± .007	.854 ± .006	.782 ± .006	.384 ± .004	.533 ± .006	.454 ± .005	.634 ± .004	.776 ± .002	.699 ± .003
	MultiKE	.411 ± .010	.521 ± .017	.468 ± .012	.495 ± .010	.646 ± .016	.569 ± .011	.290 ± .006	.357 ± .011	.326 ± .009	.319 ± .001	.396 ± .004	.360 ± .002
	RDGCN	.515 ± .008	.669 ± .006	.584 ± .007	.623 ± .006	.757 ± .004	.684 ± .005	.362 ± .002	.485 ± .002	.420 ± .002	.421 ± .001	.528 ± .002	.473 ± .002
D-Y	MTransE	.463 ± .013	.675 ± .011	.559 ± .012	.443 ± .017	.635 ± .013	.533 ± .015	.244 ± .004	.414 ± .006	.328 ± .005	.100 ± .003	.195 ± .005	.152 ± .004
	IPTransE	.313 ± .009	.456 ± .015	.378 ± .011	.752 ± .018	.873 ± .013	.808 ± .015	.396 ± .014	.558 ± .018	.474 ± .015	.456 ± .016	.620 ± .017	.534 ± .017
	JAPE	.469 ± .009	.687 ± .011	.567 ± .009	.345 ± .010	.546 ± .013	.440 ± .011	.287 ± .007	.474 ± .008	.379 ± .007	.127 ± .004	.244 ± .006	.189 ± .005
	KDCoE	.661 ± .013	.764 ± .036	.710 ± .021	.895 ± .013	.974 ± .003	.932 ± .008	.565 ± .001	.646 ± .002	.605 ± .002	.540 ± .001	.621 ± .002	.581 ± .001
	BootEA	.739 ± .014	.849 ± .010	.788 ± .012	.958 ± .001	.984 ± .001	.969 ± .001	.703 ± .004	.827 ± .003	.761 ± .003	.886 ± .003	.944 ± .002	.912 ± .002
	GCNAlign	.465 ± .012	.626 ± .011	.536 ± .011	.875 ± .005	.948 ± .004	.907 ± .004	.528 ± .003	.695 ± .005	.605 ± .004	.620 ± .006	.779 ± .006	.693 ± .006
	AttrE	.668 ± .012	.803 ± .009	.731 ± .010	.914 ± .015	.97 ± .007	.939 ± .012	.678 ± .017	.81 ± .012	.739 ± .015	.720 ± .01	.846 ± .007	.778 ± .008
	IMUSE	.392 ± .013	.571 ± .023	.473 ± .017	.899 ± .011	.949 ± .007	.922 ± .009	.536 ± .018	.700 ± .019	.613 ± .018	.629 ± .011	.774 ± .010	.696 ± .010
	SEA	.500 ± .011	.706 ± .012	.591 ± .012	.899 ± .005	.950 ± .003	.923 ± .004	.490 ± .042	.677 ± .040	.578 ± .040	.526 ± .021	.687 ± .021	.603 ± .021
	RSN4EA	.514 ± .003	.655 ± .004	.580 ± .003	.933 ± .003	.974 ± .001	.951 ± .002	.620 ± .002	.769 ± .003	.688 ± .002	.841 ± .003	.922 ± .001	.877 ± .002
	MultiKE	.903 ± .004	.939 ± .003	.920 ± .003	.856 ± .004	.908 ± .002	.881 ± .003	.884 ± .005	.920 ± .004	.901 ± .005	.853 ± .003	.896 ± .003	.874 ± .003
	RDGCN	.931 ± .004	.969 ± .003	.949 ± .003	.936 ± .003	.966 ± .001	.950 ± .002	.897 ± .001	.950 ± .001	.921 ± .001	.911 ± .002	.949 ± .002	.928 ± .002

Means ± stds. are shown. Top-3 results on each dataset are marked in red, blue and cyan, respectively. The same to the following.

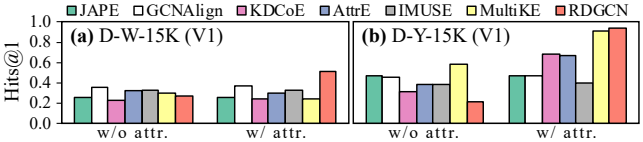


Figure 6: Hits@1 results of JAPE, GCNAlign, KDCoE, AttrE, IMUSE, MultiKE, RDGCN and their degraded variants without attribute embedding

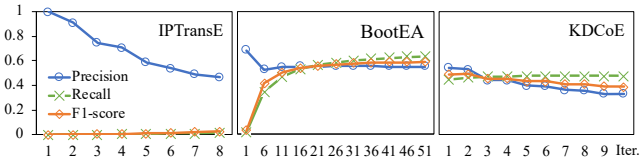


Figure 7: Precision, recall and F1-score of augmented alignment during iterations on EN-FR-100K (V1)

relations challenge many embedding approaches. Moreover, the 100K datasets have a larger candidate alignment space than the 15K datasets. It is harder to rank the target entity at the top from a larger candidate space with much more negative cases. Differently, D-Y-15K and D-Y-100K have a very similar number of relations in YAGO, which makes the results different from those on other datasets.

Relations vs. attributes. For the purely relation-based approaches, there is no clear advantage of one relation embedding technique beyond another. For example, although MTransE and BootEA both use TransE, their performance is at two extremes. We believe that the negative sampling in BootEA makes great contribution, and training embeddings only with positive samples is prone to overfitting. The work in [7] also shows that negative sampling can largely affect the expressiveness of KG embeddings. We apply the negative sampling along with the marginal ranking loss to MTransE and find that its Hits@1 on EN-FR-15K (V1) rises to 0.271, which further demonstrates the effectiveness of negative sampling. Besides, the bootstrapping strategy of BootEA also contributes a lot, which is discussed shortly. For another example, IPTransE and RSN4EA both extend triple-based embedding by linking relation triples into long relation paths, but their results are also significantly different. This is because the recurrent skipping network of RSN4EA is more powerful than the shallow composition of IPTransE.

For the approaches using attributes, we compare them to their variants without attribute embedding. Figure 6 shows

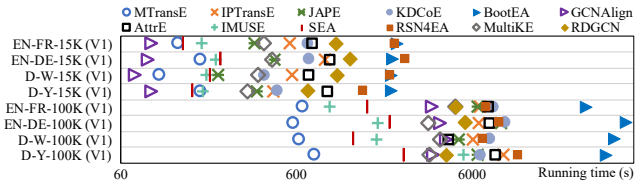


Figure 8: Running time (in log scale) on the V1 datasets

the Hits@1 results on D-W-15K (V1) and D-Y-15K (V1). Other datasets show similar results. On D-Y, we do not observe notable improvement from JAPE and GCNAlign by using attribute correlations to cluster entities. This technique would fail to capture the attribute correlations across different KGs without pre-aligned attributes. Moreover, even if the attribute correlations are discovered, this signal is too coarse-grained to determine whether two entities with correlated attributes are aligned. Differently, literal embedding brings significant improvement to most approaches except IMUSE, indicating that literals are a stronger signal for entity alignment than attribute correlations. IMUSE has a preprocessing step using literals to find new entity alignment to augment training data. However, the errors in new alignment also harm performance. Most approaches fail to be improved by attribute embedding on D-W. The symbolic heterogeneity of attributes in Wikidata (e.g., the local names of attributes are numeric IDs) notably challenges some approaches as they cannot automatically find high-quality attribute alignment for literal comparison. *Overall, attribute heterogeneity has a strong effect on capturing attribute correlations, and literal embedding facilitates entity alignment.*

Semi-supervised learning strategies. We further investigate the strengths and limitations of these semi-supervised learning strategies by analyzing the quality of the augmented seed alignment. Figure 7 depicts the precision, recall and F1-score of IPTransE, BootEA and KDCoE during the semi-supervised training on EN-FR-100K (V1), and other datasets show similar results. IPTransE fails to achieve good performance, because it involves many errors as the self-training continues but does not design a mechanism to eliminate these errors. KDCoE propagates new alignment by co-training two orthogonal types of features, i.e., relation triples and textual descriptions. However, many entities lack textual descriptions, preventing KDCoE from finding alignment seeds to augment training data. Thus, its strategy does not bring notable improvement. BootEA employs a heuristic editing method to remove wrong alignment. After undergoing a period of fluctuations, the precision stays stable while the recall continues growing during self-training, which brings a clear performance boost. We also conduct an ablation study on BootEA, and find that its self-training strategy can bring an improvement of more than 0.086 Hits@1 on the V1 datasets, demonstrating its effectiveness. *So, the quantity and quality of the augmented entity alignment have great impact on the semi-supervised approaches.* A larger augmented alignment of higher precision leads to better performance.

Running time comparison. In Figure 8, we show a brief comparison on the average running time of five repetitions on the V1 datasets. The time used by different approaches varies greatly. In general, an approach takes more time to run on a 100K dataset than on a 15K dataset. BootEA is much slower than other approaches. For example, its running time on EN-FR-15K (V1) and EN-FR-100K (V1) are 2,260 and 26,939 seconds, respectively, where the truncated negative sampling and bootstrapping procedure cost more than 23.5% and 13.3% of the time, respectively. RSN4EA also uses much time, especially on the 15K (V1) datasets, since it is trained with multi-hop paths, which are far more than the relation triples (i.e., one-hop paths). For instance, the number of two-hop paths in EN-FR-15K (V1) is 500,260, five times more than that of relation triples (88,198). As for KDCoE and AttrE, a lot of their time is spent on encoding the literal information. For example, in KDCoE, the time for training descriptions takes up at least 26.3%. By contrast, GCNAlign and MTransE use much less time, as they only use relation triples and also have a lightweight model complexity. Thus, *we recognize that using auxiliary information or techniques to boost performance usually increases training time.* Overall, MultiKE balances well between effectiveness and efficiency, as its multi-view discriminative features make it converge fast for entity alignment.

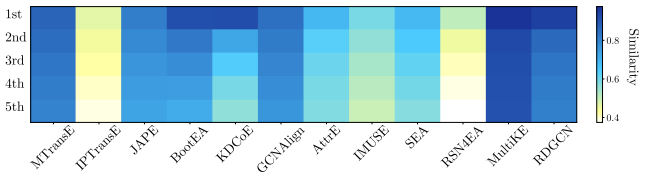


Figure 9: Visualization of the similarities between entities and their top-5 nearest cross-KG neighbors on the D-Y-15K (V1) dataset. The five rows from top to bottom correspond to the similarities from the first to the fifth nearest neighbors, respectively. Darker color indicates larger similarity.

6. EXPLORATORY EXPERIMENTS

6.1 Geometric Analysis

In addition to performance comparison, we hereby focus on the geometric properties of entity embeddings, to understand how these embeddings support entity alignment performance and the underlying limitations of existing approaches.

6.1.1 Similarity Distribution

Given entity embeddings, the alignment inference algorithm identifies aligned entities by the nearest neighbor search in the embedding space. It is interesting to investigate the similarity distribution of each entity and its nearest neighbors in the cross-KG scope. Towards this end, we visualize in Figure 9 the average similarities between entities of the source KG and their top-5 nearest neighbors of the target KG on D-Y-15K (V1). To make the similarity comparable over all the approaches, we select cosine similarity as the metric. The results show two interesting findings:

First, the average similarities between source entities and their top-1 nearest neighbors (top-1 similarities) of different approaches differ widely. BootEA, KDCoE, MultiKE and RDGCN yield a very high top-1 similarity, while IPTransE and RSN4EA show the opposite. Intuitively, a high top-1 similarity indicates a better quality because it can reflect how confidently the entity embeddings capture the alignment information between two KGs. Most approaches with a high top-1 similarity, such as BootEA, MultiKE and RDGCN, also achieve good performance for entity alignment (see Table 5). For KDCoE, as shown in Figure 7, the low precision of its augmented alignment makes its top-1 entity alignment contain many errors. Hence, its performance is not as good

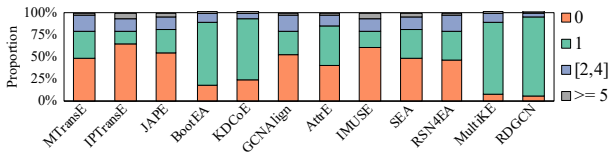


Figure 10: Proportions of target entities appearing 0, 1 and more times as the nearest neighbors on D-Y-15K (V1)

as BootEA. But it still outperforms many others as its description and relation embeddings are complementary, and thus can help find some correct alignment.

Second, the similarity variances between the top-5 nearest neighbors also differ greatly, which can be reflected by the color gradients of the five rows from top to bottom. BootEA, KDCoE, RSN4EA and RDGCN exhibit large variances while MTransE, IPTransE and JAPE exhibit very slight variances. A small similarity variance means that the nearest neighbors are not discriminative enough to enable the entity to identify its counterpart correctly. The overfitting issue in MTransE, the bootstrapping errors in IPTransE and the fuzzy entity clusters in JAPE are the reasons for their non-discriminative embeddings. Other datasets show similar distribution. *The ideal similarity distribution for entity alignment is to hold a high top-1 similarity and a large similarity variance.*

6.1.2 Hubness and Isolation

Hubness is a common phenomenon in high-dimensional vector spaces [60], where some points (known as *hubs*) frequently appear as the top-1 nearest neighbors of many other points in the vector space. Another phenomenon is that, there would exist some outliers isolated from any point clusters. The two issues have negative effects on the tasks relying on the nearest neighbor search [11, 55]. Here, we investigate whether embedding-based entity alignment also suffers from them. We measure the proportions of target entities that appear zero, one and more times as the nearest neighbors of source entities, respectively. Figure 10 shows the results on D-Y-15K (V1), and other datasets also show similar results. Surprisingly, we find that there is a large proportion of target entities that never appear as the top-1 nearest neighbors of any source entity (marked with orange bars). This means that such isolated entities may never be considered if we use the greedy strategy of choosing the top-1 nearest neighbor to form alignment. Consequently, we would miss much correct entity alignment. The entities (blue and gray bars) that appear as the nearest neighbors of more than one source entity also occupy considerable proportions. They would cause many violations against the 1-to-1 mapping constraint and globally increase the uncertainty of alignment inference. We observe that the approaches which yield fewer isolated and hub entities, such as MultiKE and RDGCN, achieve the leading performance of entity alignment, and vice versa. So, the ideal case is to have small proportions of isolated and hub entities. *This finding indicates that we can make an estimation about the final entity alignment performance through the hubness and isolation analysis.*

To resolve the hubness and isolation problem, we explore cross-domain similarity local scaling (CSLS) [11] as the alternative metric. It normalizes the similarity of source and target entity embeddings based on the density of their embedding neighbors. Taking cosine for example, we have

$$\text{CSLS}(\mathbf{x}_s, \mathbf{x}_t) = 2\cos(\mathbf{x}_s, \mathbf{x}_t) - \psi_t(\mathbf{x}_s) - \psi_s(\mathbf{x}_t), \quad (7)$$

Table 6: Hits@1 w.r.t. distance metrics and alignment inference strategies on D-Y-15K (V1)

	Greedy	Greedy w/ CSLS	SM	SM w/ CSLS
MTransE	.463 ± .013	.550 ± .009	.694 ± .006	.697 ± .010
IPTransE	.313 ± .009	.339 ± .013	.370 ± .018	.369 ± .018
JAPE	.469 ± .009	.549 ± .009	.692 ± .015	.691 ± .015
KDCoE	.661 ± .013	.679 ± .000	.840 ± .024	.815 ± .031
BootEA	.739 ± .014	.741 ± .009	.783 ± .007	.782 ± .006
GCNAlign	.465 ± .012	.531 ± .008	.613 ± .008	.582 ± .010
AttrE	.668 ± .012	.778 ± .012	.845 ± .012	.857 ± .012
IMUSE	.392 ± .013	.448 ± .018	.520 ± .028	.518 ± .030
SEA	.500 ± .011	.557 ± .017	.647 ± .012	.650 ± .012
RSN4EA	.514 ± .003	.548 ± .003	.571 ± .002	.575 ± .004
MultiKE	.903 ± .004	.925 ± .003	.951 ± .003	.956 ± .002
RDGCN	.931 ± .004	.956 ± .002	.962 ± .002	.979 ± .001

where $\psi_t(\mathbf{x}_s)$ denotes the average similarity between the source entity $\mathbf{x}_s$ and its top-k nearest neighbors in the target KG. $\psi_s(\mathbf{x}_t)$ is computed symmetrically. CSLS decreases the similarities between hub entities and other entities. It can also let some isolated entities be fairly considered in testing because they usually receive less similarity penalization. Therefore, we use CSLS to enhance the conventional distance metrics. In addition, we also consider stable matching (a.k.a. stable marriage) to retrieve entity alignment from a global perspective rather than the greedy strategy based on the nearest neighbor search. The entity alignment between two KGs is stable when there does not exist another predicted aligned pair (e_1, e_2) of higher preference than those of e_1 and e_2 to their current matches. The preference can be calculated based on a similarity metric such as CSLS.

We report the Hits@1 results enhanced by CSLS and stable matching (abbr. SM) in Table 6. We find that CSLS brings significant gains to the greedy strategy, especially on MTransE, JAPE, GCNAlign and AttrE. This is because CSLS can help mitigate the hubness phenomenon. Besides, SM further brings improvement. For example, compared with the greedy strategy, it yields a gain of more than 10% on Hits@1 for MTransE, JAPE, KDCoE, GCNAlign, AttrE, IMUSE, SEA and RotatE. The reason lies in that SM can consider all entities including isolated ones. Interestingly, we observe that CSLS does not boost the performance of SM. This indicates that SM relies less on the distance metric. We gain similar results on other datasets. *In summary, existing approaches concentrate on developing more powerful embedding and interaction methods, but some methods for the alignment module can also improve performance.*

6.2 Unexplored KG Embedding Models

As summarized in Sect. 2.1, most existing approaches use TransE [5] or GCNs [38] for KG embedding due to their strong robustness and good generalizability. However, many other KG embedding models have not been explored for entity alignment yet. To fill this gap, we evaluate three translational models TransH [82], TransR [49] and TransD [33], two deep models ProjE [66] and ConvE [13], as well as three semantic matching models HolE [54], SimplE [36] and RotatE [71], for entity alignment. We choose MTransE as baseline and replace its relation embedding model TransE with the aforementioned models. We report the Hits@1 results on the V1 datasets in Figure 11. Other results are available online. The results of TransR and HolE are omitted because their Hits@1 scores are smaller than 0.01 on most datasets.

We can see that the improved translational models TransH and TransD show stable and promising performance on all the datasets. Specifically, on the 100K datasets, TransH

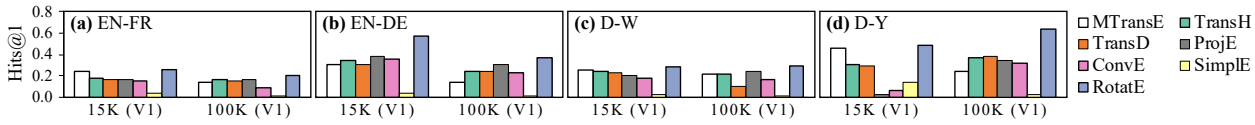


Figure 11: Cross-validation results of unexplored KG embedding models on the 15K (V1) and 100K (V1) datasets

Table 7: Comparison with conventional approaches on the 15K and 100K datasets

		15K (V1)			15K (V2)			100K (V1)			100K (V2)		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
EN-FR	LogMap	.818 $_{\pm .002}$	.729 $_{\pm .002}$	.771 $_{\pm .001}$	.599 $_{\pm .003}$	.805 $_{\pm .001}$	.687 $_{\pm .002}$	.719 $_{\pm .001}$	.677 $_{\pm .001}$	.697 $_{\pm .001}$	.541 $_{\pm .001}$	.709 $_{\pm .000}$	.614 $_{\pm .001}$
	PARIS	.907 $_{\pm .000}$	.900 $_{\pm .000}$	.903 $_{\pm .000}$	.929 $_{\pm .001}$	.939 $_{\pm .001}$	.934 $_{\pm .001}$	.852 $_{\pm .000}$	.844 $_{\pm .000}$	.848 $_{\pm .000}$	.874 $_{\pm .000}$	.888 $_{\pm .000}$	.881 $_{\pm .000}$
	OpenEA	.755 $_{\pm .004}^{*}$	.755 $_{\pm .004}^{*}$	.755 $_{\pm .004}^{*}$	.864 $_{\pm .007}^{\ddagger}$	.864 $_{\pm .007}^{\ddagger}$	.864 $_{\pm .007}^{\ddagger}$	.640 $_{\pm .004}^{*}$	.640 $_{\pm .004}^{*}$	.640 $_{\pm .004}^{*}$	.715 $_{\pm .003}^{*}$	.715 $_{\pm .003}^{*}$	.715 $_{\pm .003}^{*}$
EN-DE	LogMap	.925 $_{\pm .002}$	.725 $_{\pm .001}$	.813 $_{\pm .001}$	.888 $_{\pm .000}$	.702 $_{\pm .002}$	.784 $_{\pm .001}$	.827 $_{\pm .001}$	.732 $_{\pm .001}$	.777 $_{\pm .001}$	.729 $_{\pm .003}$	.729 $_{\pm .001}$	.729 $_{\pm .002}$
	PARIS	.938 $_{\pm .000}$	.933 $_{\pm .000}$	.935 $_{\pm .000}$	.959 $_{\pm .000}$	.964 $_{\pm .000}$	.961 $_{\pm .000}$	.890 $_{\pm .000}$	.886 $_{\pm .000}$	.888 $_{\pm .000}$	.913 $_{\pm .000}$	.923 $_{\pm .000}$	.918 $_{\pm .000}$
	OpenEA	.830 $_{\pm .006}^{*}$	.830 $_{\pm .006}^{*}$	.830 $_{\pm .006}^{*}$	.833 $_{\pm .015}^{\dagger}$	.833 $_{\pm .015}^{\dagger}$	.833 $_{\pm .015}^{\dagger}$	.722 $_{\pm .002}^{*}$	.722 $_{\pm .002}^{*}$	.722 $_{\pm .002}^{*}$	.766 $_{\pm .002}^{*}$	.766 $_{\pm .002}^{*}$	.766 $_{\pm .002}^{*}$
D-W	LogMap	-	-	-	-	-	-	-	-	-	-	-	-
	PARIS	.746 $_{\pm .001}$	.723 $_{\pm .002}$	.734 $_{\pm .001}$	.828 $_{\pm .000}$	.854 $_{\pm .001}$	.840 $_{\pm .001}$	.692 $_{\pm .000}$	.643 $_{\pm .000}$	.667 $_{\pm .000}$	.783 $_{\pm .002}$	.806 $_{\pm .002}$	.795 $_{\pm .002}$
	OpenEA	.572 $_{\pm .008}^{\dagger}$	.572 $_{\pm .008}^{\dagger}$	.572 $_{\pm .008}^{\dagger}$	.821 $_{\pm .004}^{\dagger}$	.821 $_{\pm .004}^{\dagger}$	.821 $_{\pm .004}^{\dagger}$	.516 $_{\pm .006}^{\dagger}$	.516 $_{\pm .006}^{\dagger}$	.516 $_{\pm .006}^{\dagger}$	.766 $_{\pm .007}^{\dagger}$	.766 $_{\pm .007}^{\dagger}$	.766 $_{\pm .007}^{\dagger}$
D-Y	LogMap	.971 $_{\pm .001}$	.943 $_{\pm .002}$	.957 $_{\pm .001}$	.944 $_{\pm .002}$	.984 $_{\pm .001}$	.964 $_{\pm .002}$	.920 $_{\pm .001}$	.915 $_{\pm .001}$	.917 $_{\pm .001}$	.954 $_{\pm .001}$	.912 $_{\pm .000}$	.933 $_{\pm .001}$
	PARIS	.899 $_{\pm .002}$	.869 $_{\pm .002}$	.884 $_{\pm .002}$	.972 $_{\pm .000}$	.971 $_{\pm .000}$	.971 $_{\pm .000}$	.893 $_{\pm .000}$	.866 $_{\pm .000}$	.880 $_{\pm .000}$	.956 $_{\pm .000}$	.953 $_{\pm .000}$	.955 $_{\pm .000}$
	OpenEA	.931 $_{\pm .004}^{*}$	.931 $_{\pm .004}^{*}$	.931 $_{\pm .004}^{*}$	.958 $_{\pm .001}^{\dagger}$	.958 $_{\pm .001}^{\dagger}$	.958 $_{\pm .001}^{\dagger}$	.897 $_{\pm .001}^{*}$	.897 $_{\pm .001}^{*}$	.897 $_{\pm .001}^{*}$	.911 $_{\pm .002}^{*}$	.911 $_{\pm .002}^{*}$	.911 $_{\pm .002}^{*}$

The results of RDGCN, BootEA and MultiKE are marked in "$*$", "$\dagger$" and "$\ddagger$", respectively.

is robuster than MTransE and gain better results. This is because TransH handles multi-mapping relations better and also uses negative sampling to enhance embedding. Differently, we find that TransR fails to achieve promising results. The relation-specific transformation of entity embeddings in TransR requires relation alignment to propagate the alignment information between entities. However, in our problem setting, we focus on entity alignment and do not provide relation alignment due to the great heterogeneity between KG schemata. The neural models ConvE and ProjE also show promising results on most of our datasets. However, we find that they perform poorly on D-Y-15K (V1). We owe it to the fewer relation triples and the big gap between the relation numbers in these datasets. It is difficult for the two-dimensional convolution of ConvE or the non-linear transformation of ProjE to capture the similar interactions between entity and relation embeddings across such heterogeneous KGs. For the semantic matching models, the non-Euclidean embedding model RotatE achieves much better performance than SimplE. It also outperforms other models. *In short, not all KG embedding models are suitable for entity alignment, and non-Euclidean embeddings are worth further exploration.*

6.3 Comparison to Conventional Approaches

We compare OpenEA with two famous open-source conventional approaches for KG alignment, i.e., LogMap [34] from the Semantic Web community and PARIS [70] from the Database community. LogMap is an ontology matching system with built-in reasoning and inconsistency repair capabilities. PARIS is a holistic solution to align KGs based on probability estimates. The non-English KGs in cross-lingual datasets are translated to English using Google Translate to eliminate the language barrier for LogMap and PARIS.

Overall comparison. Table 7 compares LogMap, PARIS and the best embedding-based approach in OpenEA. For the test phase of OpenEA, as each source entity gets a list of candidates, precision, recall and F1-score are in fact equal to Hits@1. All these approaches achieve good results, where PARIS performs the best on most of our datasets including EN-FR, EN-DE and D-W, and LogMap achieves promising performance on D-Y. Overall, OpenEA shows no superiority over the conventional approaches PARIS and LogMap. We think this is because current embedding-based approaches put in their main efforts in learning expressive embeddings to capture entity features while ignore the alignment inference. As summarized in Sect. 2.2.2, their alignment inference strategies are based on pairwise similarity comparison, lacking the capability of inconsistency repair and holistic estimation that LogMap and PARIS have. Our geometric analysis in Sect. 6.1 further shows that this weakness would lead to the issue of hubness and isolation and thus degrade entity alignment performance. By resolving this issue, as shown in Table 6, OpenEA (RDGCN) achieves better Hits@1 (precision) on D-Y-15K (V1) and outperforms LogMap and PARIS in Table 7. Our experiment indicates that embedding-based entity alignment approaches require further improvement in alignment inference. Besides, we notice that LogMap fails to output entity alignment on the D-W datasets. This is because LogMap highly depends on the local names in URIs to compute similarities while the URIs in Wikidata have no actual meanings (e.g., `https://www.wikidata.org/wiki/Property:P69`). In fact, the results of all approaches on D-W decrease severely. The symbolic heterogeneity brings huge obstacles to both conventional and embedding-based approaches.

Table 8: Comparison with conventional approaches using different features on EN-FR-15K (V1)

	Using relation triples only			Using attribute triples only		
	Precision	Recall	F1-score	Precision	Recall	F1-score
LogMap	-	-	-	.816 $_{\pm .003}$	.723 $_{\pm .002}$	.767 $_{\pm .001}$
PARIS	-	-	-	.917 $_{\pm .000}$	.769 $_{\pm .000}$	.837 $_{\pm .000}$
BootEA	.507 $_{\pm .010}$	.507 $_{\pm .010}$	.507 $_{\pm .010}$	-	-	-
MultiKE	.337 $_{\pm .005}$	.337 $_{\pm .005}$	.337 $_{\pm .005}$	.719 $_{\pm .005}$	.719 $_{\pm .005}$	.719 $_{\pm .005}$
RDGCN	.255 $_{\pm .004}$	.255 $_{\pm .004}$	.255 $_{\pm .004}$	-	-	-

Feature study. Table 8 shows the results of LogMap and PARIS as well as three top-performing embedding-based approaches RDGCN, BootEA and MultiKE when only given relation or attribute triples of EN-FR-15K (V1). *LogMap and PARIS rely on attribute triples and fail to output alignment in the case of using relation triples only*. This is different from embedding-based approaches that all use relation triples. In the case of using relation triples only, BootEA is not affected by the lack of attribute triples. The performance of MultiKE and RDGCN drops greatly since their attribute embedding modules are disabled in this case. However, their relation embedding modules can still learn embeddings. When only using attribute triples, the results of LogMap almost remain intact because it mainly uses attribute triples to compute entity similarities. The recall of PARIS drops dramatically as it cannot use the relational inference to find more entity

Table 9: Summary of the required information of embedding-based and conventional entity alignment approaches

	MTransE	IPTransE	JAPE	KDCoE	BootEA	GCNAlign	AttrE	IMUSE	SEA	RSN4EA	MultiKE	RDGCN	LogMap	PARIS
Relation/attribute triples	∗/	∗/	∗/○	○/○	∗/	∗/○	○/○	○/○	∗/	∗/	○/○	∗/○	○/∗	○/∗
Pre-aligned ent./prop.	∗/○	∗/○	∗/○	∗/	∗/	∗/○	∗/	∗/	∗/	∗/	∗/○	∗/	○/○	○/○
Word embed./Google trans.				○/			○/				○/	○/	/△	/△

"∗" means "mandatory", "○" means "optional", "△" means "mandatory for cross-lingual entity alignment", and blank means "not applicable".

alignment. But its precision is still very high and even a little better than that in Table 7. Considering that PARIS is not designed for relational inference, relation triples may bring along noises to this approach. As for embedding-based approaches, RDGCN and BootEA cannot learn embeddings without relation triples. The multi-view approach MultiKE also suffers from performance penalties because it cannot benefit from the relation embedding. This experiment reveals the different application scenarios of these entity alignment approaches. *Conventional approaches better support the entity alignment scenario with attribute information. Embedding-based approaches cover most of the typical scenarios with either relation information, attribute information or both.*

Analysis on predicted alignment. To further investigate the potential complementarity of embedding-based and conventional approaches, we show in Figure 12 the proportions of correct alignment found by OpenEA (RDGCN), LogMap and PARIS on EN-FR-100K (V1). They all suffer from the same challenge (the symbolic heterogeneity). We find that they can produce complementary entity alignment. This analysis calls for a hybrid system for entity alignment built on both conventional and embedding-based techniques.

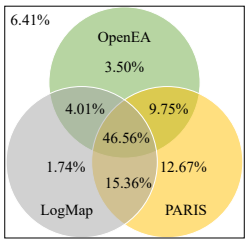


Figure 12: Proportions of the correct alignment found by LogMap, PARIS and OpenEA on EN-FR-100K (V1). OpenEA additionally finds 13.25% (3.50% + 9.75%) and 7.51% (3.50% + 4.01%) of the alignment that LogMap and PARIS do not, respectively. Besides, 6.41% of the alignment is not found by any approach, while 45.56% is found by all the three.

7. SUMMARY AND FUTURE DIRECTIONS

7.1 Summary of Experiments

From our experimental results, we find that (i) RDGCN, BootEA and MultiKE achieve the most competitive performance. This suggests that incorporating both literal information and carefully-designed bootstrapping can help entity alignment. (ii) For the embedding models designed for link prediction, we find that not all of them are suitable for entity alignment. (iii) Currently, the alignment inference strategy receives little attention. Our preliminary results show that the CSLS distance metric and the stable matching strategy can bring performance improvement to all the approaches. (iv) We also find that embedding-based and conventional entity alignment approaches are complementary to each other. (v) For choosing appropriate approaches based on the available resources in real-world scenarios, Table 9 summarizes the required information of embedding-based and conventional entity alignment approaches in our experimental analysis.

7.2 Future Directions

Unsupervised entity alignment. As summarized in Section 2.2.3 and discussed in Section 5.2, all current approaches require seed alignment as supervision. However, this requirement is sometimes difficult to be satisfied in the real world. Hence, studying unsupervised entity alignment is a meaningful direction. A possible solution is to incorporate auxiliary features or resources and distill distant supervision from them, such as discriminative features (homepages of people and introductory images of products) and pre-trained word embeddings [87]. Besides, recent advances in unsupervised cross-lingual word alignment [11] like orthogonal Procrustes [64] and adversarial training [23] are also worth investigation. Another possible solution is to use active learning [32, 59] or abductive learning [92] to reduce the burden of data labeling.

Long-tail entity alignment. Our experimental analysis on the sparse and dense datasets reveals the difficulty in aligning long-tail entities, which usually account for a large proportion in KGs [43]. To embed long-tail entities, in addition to using more advanced graph neural networks [38, 63, 78], injecting more features such as multi-modal data and taxonomies would also be helpful. As KGs are far from complete, jointly training link prediction and entity alignment via a unified framework may leverage the incidental supervision of both tasks. Extracting additional information from the open Web to enrich long-tail entities is also a potential direction [67].

Large-scale entity alignment. The running time comparison shows that training existing approaches on larger datasets costs much more time. The test phrase also takes much time. For example, computing the pairwise cosine similarity of entity embeddings on a 100K dataset costs about 8 minutes by using 10 processes in parallel. The cost would grow polynomially along with the growing number of entities. It is difficult for embedding-based (and also conventional) approaches to run on very large KGs due to the large and unpartitioned candidate space. The *blocking* techniques, e.g., locality-sensitive hashing [21] and hashing representation learning [45], may be useful to narrow the candidate space.

Entity alignment in non-Euclidean spaces. Our experimental results in Figure 11 indicate that the non-Euclidean embedding model RotatE [71] outperforms other Euclidean models. We also notice that recent non-Euclidean embeddings have demonstrated their effectiveness in representing graph-structured data [53]. So, alignment-oriented non-Euclidean KG embedding models are worth exploiting.

8. CONCLUSION

In this paper, we survey the field of embedding-based entity alignment between KGs and conduct a benchmarking study of the representative approaches. We create a set of dedicated datasets that better fit real-world KGs and develop an open-source library containing a variety of entity alignment approaches and KG embedding models. Our experiments analyze the status quo and point out future directions.

Acknowledgments. This work was supported by the National Key R&D Program of China (No. 2018YFB1004300), the National Natural Science Foundation of China (No. 61872172), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

9. REFERENCES

[1] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *SIGMOD*, pages 1995–2010. ACM, 2020.

[2] B. Bahmani, A. Chowdhury, and A. Goel. Fast incremental and personalized pagerank. *PVLDB*, 4(3):173–184, 2010.

[3] I. Balažević, C. Allen, and T. M. Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *EMNLP-IJCNLP*, pages 5184–5193, Hong Kong, China, 2019. ACL.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, Lake Tahoe, NV, USA, 2013. Curran Associates, Inc.

[6] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, Banff, Canada, 2014.

[7] L. Cai and W. Y. Wang. KBGAN: Adversarial learning for knowledge graph embeddings. In *NAACL-HLT*, pages 1470–1480, New Orleans, LA, USA, 2018. ACL.

[8] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li, and T. Chua. Multi-channel graph neural network for entity alignment. In *ACL*, pages 1452–1461, Florence, Italy, 2019. ACL.

[9] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI*, pages 3998–4004, Stockholm, Sweden, 2018. IJCAI Organization.

[10] M. Chen, Y. Tian, M. Yang, and C. Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517, Melbourne, Australia, 2017. IJCAI Organization.

[11] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. In *ICLR*, Vancouver, Canada, 2018.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, Barcelona, Spain, 2016. Curran Associates, Inc.

[13] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. Convolutional 2D knowledge graph embeddings. In *AAAI*, pages 1811–1818, New Orleans, LA, USA, 2018. AAAI Press.

[14] M. Ebraheem, S. Thirumuruganathan, S. R. Joty, M. Ouzzani, and N. Tang. Distributed representations of tuples for entity resolution. *PVLDB*, 11(11):1454–1467, 2018.

[15] A. El-Roby and A. Aboulnaga. ALEX: Automatic link exploration in linked data. In *SIGMOD*, pages 1839–1853, Melbourne, Australia, 2015. ACM.

[16] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):1–16, 2007.

[17] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[18] Y. Gao, J. Liang, B. Han, M. Yakout, and A. Mohamed. Building a large-scale, accurate and fresh knowledge graph. `https://kdd2018tutorialt39.azurewebsites.net/`, 2018.

[19] W. Ge, J. Chen, W. Hu, and Y. Qu. Object link structure in the semantic web. In *ESWC*, pages 257–271, Heraklion, Greece, 2010. Springer.

[20] L. Getoor and A. Machanavajjhala. Entity resolution: Tutorial. `http://users.umiacs.umd.edu/~getoor/Tutorials/ER_VLDB2012.pdf`, 2012.

[21] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, Edinburgh, Scotland, 1999.

[22] H. Glaser, A. Jaffri, and I. Millard. Managing co-reference on the semantic web. In *WWW Workshop on Linked Data on the Web*, Madrid, Spain, 2009.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, Montréal, Canada, 2014. Curran Associates, Inc.

[24] L. Guo, Z. Sun, and W. Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML*, pages 2505–2514, Long Beach, CA, USA, 2019. PMLR.

[25] L. Guo, Q. Zhang, W. Hu, Z. Sun, and Y. Qu. Learning to complete knowledge graphs with deep sequential models. *Data Intelligence*, 1(3):289–308, 2019.

[26] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74, 2017.

[27] X. Han, S. Cao, X. Lv, Y. Lin, Z. Liu, M. Sun, and J. Li. OpenKE: An open toolkit for knowledge embedding. In *EMNLP (demonstration)*, pages 139–144, Brussels, Belgium, 2018. ACL.

[28] F. He, Z. Li, Y. Qiang, A. Liu, G. Liu, P. Zhao, L. Zhao, M. Zhang, and Z. Chen. Unsupervised entity alignment using attribute triples and relation triples. In *DASFAA*, pages 367–382, Chiang Mai, Thailand, 2019. Springer.

[29] M. Heimann, H. Shen, T. Safavi, and D. Koutra. REGAL: Representation learning-based graph alignment. In *CIKM*, pages 117–126, Torino, Italy, 2018. ACM.

[30] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Supercomputing*, page 28, San Diego, CA, USA, 1995. ACM/IEEE.

[31] W. Hu, J. Chen, and Y. Qu. A self-training approach for resolving object coreference on the semantic web. In *WWW*, pages 87–96, Hyderabad, India, 2011. ACM.

[32] R. Isele and C. Bizer. Learning expressive linkage rules using genetic programming. *PVLDB*, 5(11):1638–1649, 2012.

[33] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao. Knowledge

graph embedding via dynamic mapping matrix. In *ACL*, pages 687–696, Beijing, China, 2015. ACL.

[34] E. Jiménez-Ruiz and B. C. Grau. LogMap: Logic-based and scalable ontology matching. In *ISWC*, volume LNCS 7031, pages 273–288, Bonn, Germany, 2011. Springer.

[35] R. Kadlec, O. Bajgar, and J. Kleindienst. Knowledge base completion: Baselines strike back. In *ACL Workshop on Representation Learning for NLP*, pages 69–74, Vancouver, Canada, 2017. ACL.

[36] S. M. Kazemi and D. Poole. Simple embedding for link prediction in knowledge graphs. In *NeurIPS*, pages 4289–4300, Montréal, Canada, 2018. Curran Associates, Inc.

[37] A. Kimmig, A. Memory, R. J. Miller, and L. Getoor. A collective, probabilistic approach to schema mapping. In *ICDE*, pages 921–932, San Diego, CA, USA, 2017. IEEE.

[38] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, Toulon, France, 2017.

[39] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani. SiGMa: Simple greedy matching for aligning large knowledge bases. In *KDD*, pages 572–580, Chicago, IL, USA, 2013. ACM.

[40] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[41] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *KDD*, pages 631–636, Philadelphia, PA, USA, 2006. ACM.

[42] C. Li, Y. Cao, L. Hou, J. Shi, J. Li, and T. Chua. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *EMNLP-IJCNLP*, pages 2723–2732, Hong Kong, China, 2019. ACL.

[43] F. Li, X. L. Dong, A. Langen, and Y. Li. Knowledge verification for long-tail verticals. *PVLDB*, 10(11):1370–1381, 2017.

[44] S. Li, X. Li, R. Ye, M. Wang, H. Su, and Y. Ou. Non-translational alignment for multi-relational networks. In *IJCAI*, pages 4180–4186, Stockholm, Sweden, 2018. IJCAI Organization.

[45] D. Lian, K. Zheng, V. W. Zheng, Y. Ge, L. Cao, I. W. Tsang, and X. Xie. High-order proximity preserving information network hashing. In *KDD*, pages 1744–1753, London, UK, 2018. ACM.

[46] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[47] X. Lin, H. Yang, J. Wu, C. Zhou, and B. Wang. Guiding cross-lingual entity alignment via adversarial knowledge embedding. In *ICDM*, pages 429–438, Beijing, China, 2019. IEEE.

[48] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun. Knowledge representation learning: A quantitative review. *CoRR*, page abs/1812.10901, 2018.

[49] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, Austin, TX, USA, 2015. AAAI Press.

[50] D. G. McVitie and L. B. Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486–490, 1971.

[51] I. Megdiche, O. Teste, and C. T. dos Santos. An extensible linear approach for holistic ontology matching. In *ISWC*, volume LNCS 9981, pages 393–410, Kobe, Japan, 2016. Springer.

[52] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra. Deep learning for entity matching: A design space exploration. In *SIGMOD*, pages 19–34, Houston, TX, USA, 2018. ACM.

[53] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6341–6350, Long Beach, CA, USA, 2017. Curran Associates, Inc.

[54] M. Nickel, L. Rosasco, and T. A. Poggio. Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961, Phoenix, AZ, USA, 2016. AAAI Press.

[55] A. Ormazabal, M. Artetxe, G. Labaka, A. Soroa, and E. Agirre. Analyzing the limitations of cross-lingual word embedding mappings. In *ACL*, pages 4990–4995, Florence, Italy, 2019. ACL.

[56] N. Otani, H. Kiyomaru, D. Kawahara, and S. Kurohashi. Cross-lingual knowledge projection using machine translation and target-side knowledge base completion. In *COLING*, pages 1508–1520, Santa Fe, NM, USA, 2018. ACL.

[57] S. Pei, L. Yu, R. Hoehndorf, and X. Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *WWW*, pages 3130–3136, San Francisco, CA, USA, 2019. ACM.

[58] S. Pei, L. Yu, and X. Zhang. Improving cross-lingual entity alignment via optimal transport. In *IJCAI*, pages 3231–3237, Macao, China, 2019. IJCAI Organization.

[59] K. Qian, L. Popa, and P. Sen. Active learning for large-scale entity resolution. In *CIKM*, pages 1379–1388, Singapore, 2017. ACM.

[60] M. Radovanović, A. Nanopoulos, and M. Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

[61] T. Rebele, F. M. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *ISWC*, volume LNCS 9982, pages 177–185, Kobe, Japan, 2016. Springer.

[62] A. Rossi, D. Firmani, A. Matinata, P. Merialdo, and D. Barbosa. Knowledge graph embedding for link prediction: A comparative analysis. *CoRR*, page abs/2002.00819, 2020.

[63] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *ESWC*, volume LNCS 10843, pages 593–607, Heraklion, Greece, 2018. Springer.

[64] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[65] C. Shao, L.-M. Hu, J.-Z. Li, Z.-C. Wang, T. Chung, and J.-B. Xia. RiMOM-IM: A novel iterative framework for instance matching. *Journal of Computer Science and Technology*, 31(1):185–197, 2016.

[66] B. Shi and T. Weninger. ProjE: Embedding projection for knowledge graph completion. In *AAAI*, pages 1236–1242, San Francisco, CA, USA, 2017. AAAI Press.

[67] B. Shi and T. Weninger. Open-world knowledge graph completion. In *AAAI*, pages 1957–1964. AAAI Press, 2018.

[68] X. Shi and Y. Xiao. Modeling multi-mapping relations for precise cross-lingual entity alignment. In *EMNLP-IJCNLP*, pages 813–822, Hong Kong, China, 2019. ACL.

[69] M. P. H. Stumpf, C. Wiuf, and R. M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

[70] F. M. Suchanek, S. Abiteboul, and P. Senellart. PARIS: Probabilistic alignment of relations, instances, and schema. *PVLDB*, 5(3):157–168, 2012.

[71] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *ICLR*, New Orleans, LA, USA, 2019.

[72] Z. Sun, W. Hu, and C. Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, volume LNCS 10587, pages 628–644, Vienna, Austria, 2017. Springer.

[73] Z. Sun, W. Hu, Q. Zhang, and Y. Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402, Stockholm, Sweden, 2018. IJCAI Organization.

[74] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*, pages 222–229, New York City, NY, USA, 2020. AAAI Press.

[75] K. Toutanova and D. Chen. Observed versus latent features for knowledge base and text inference. In *CVSC*, pages 57–66, Beijing, China, 2015. ACL.

[76] T. Trouillon, J. Welbl, S. Riedel, Éric Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080, New York City, NY, USA, 2016. PMLR.

[77] B. D. Trsedya, J. Qi, and R. Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI*, pages 297–304, Honolulu, HI, USA, 2019. AAAI Press.

[78] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR*, Vancouver, Canada, 2018.

[79] D. Vrandečič and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[80] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[81] Z. Wang, Q. Lv, X. Lan, and Y. Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357, Brussels, Belgium, 2018. ACL.

[82] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119, Québec City, Canada, 2014. AAAI Press.

[83] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*, pages 5278–5284, Macao, China, 2019. IJCAI Organization.

[84] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao. Jointly learning entity and relation representations for entity alignment. In *EMNLP-IJCNLP*, pages 240–249, Hong Kong, China, 2019. ACL.

[85] K. Xu, L. Wang, M. Yu, Y. Feng, Y. Song, Z. Wang, and D. Yu. Cross-lingual knowledge graph alignment via graph matching neural network. In *ACL*, Florence, Italy, 2019. ACL.

[86] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, San Diego, CA, USA, 2015.

[87] H. Yang, Y. Zou, P. Shi, W. Lu, J. Lin, and X. Sun. Aligning cross-lingual entities with multi-aspect information. In *EMNLP-IJCNLP*, pages 4430–4440, Hong Kong, China, 2019. ACL.

[88] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang. A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI*, pages 4135–4141, Macao, China, 2019. IJCAI Organization.

[89] J. Zhang, B. Chen, X. Wang, H. Chen, C. Li, F. Jin, G. Song, and Y. Zhang. MEgo2Vec: Embedding matched ego networks for user alignment across social networks. In *CIKM*, pages 327–336, Turin, Italy, 2018. ACM.

[90] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu. Multi-view knowledge graph embedding for entity alignment. In *IJCAI*, pages 5429–5435, Macao, China, 2019. IJCAI Organization.

[91] X. Zhou, Q. Zhu, P. Liu, and L. Guo. Learning knowledge embeddings by combining limit-based scoring loss. In *CIKM*, pages 1009–1018, Singapore, 2017. ACM.

[92] Z.-H. Zhou. Abductive learning: Towards bridging machine learning and logical reasoning. *SCIENCE CHINA Information Sciences*, 62(7):76101:1–76101:3, 2019.

[93] H. Zhu, R. Xie, Z. Liu, and M. Sun. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, pages 4258–4264, Melbourne, Australia, 2017. IJCAI Organization.

[94] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *IJCAI*, pages 1943–1949, Macao, China, 2019. IJCAI Organization.

[95] Y. Zhuang, G. Li, Z. Zhong, and J. Feng. PBA: Partition and blocking based alignment for large knowledge bases. In *DASFAA*, pages 415–431, Dallas, TX, USA, 2016. Springer.

[96] Y. Zhuang, G. Li, Z. Zhong, and J. Feng. Hike: A hybrid human-machine method for entity alignment in large-scale knowledge bases. In *CIKM*, pages 1917–1926, Singapore, 2017. ACM.