Reinforcement Learning-based Collective Entity Alignment with Adaptive Features

Zeng, W.; Zhao, X.; Tang, J.; Lin, X.; Groth, P.

DOI
10.1145/3446428

Publication date
2021

Document Version
Final published version

Published in
ACM Transactions on Information Systems

License
Unspecified

Reinforcement Learning–based Collective Entity Alignment with Adaptive Features

WEIXIN ZENG, XIANG ZHAO, and JIUYANG TANG, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, China
XUEMIN LIN, The University of New South Wales, Australia
PAUL GROTH, University of Amsterdam, The Netherlands

Entity alignment (EA) is the task of identifying the entities that refer to the same real-world object but are located in different knowledge graphs (KGs). For entities to be aligned, existing EA solutions treat them separately and generate alignment results as ranked lists of entities on the other side. Nevertheless, this decision-making paradigm fails to take into account the interdependence among entities. Although some recent efforts mitigate this issue by imposing the 1-to-1 constraint on the alignment process, they still cannot adequately model the underlying interdependence and the results tend to be sub-optimal.

To fill in this gap, in this work, we delve into the dynamics of the decision-making process, and offer a reinforcement learning (RL)–based model to align entities collectively. Under the RL framework, we devise the coherence and exclusiveness constraints to characterize the interdependence and restrict collective alignment. Additionally, to generate more precise inputs to the RL framework, we employ representative features to capture different aspects of the similarity between entities in heterogeneous KGs, which are integrated by an adaptive feature fusion strategy. Our proposal is evaluated on both cross-lingual and mono-lingual EA benchmarks and compared against state-of-the-art solutions. The empirical results verify its effectiveness and superiority.

CCS Concepts: • **Information systems → Information integration**; **Data extraction and integration**;

Additional Key Words and Phrases: Entity alignment, reinforcement learning, adaptive feature fusion

ACM Reference format:
Weixin Zeng, Xiang Zhao, Jiuyang Tang, Xuemin Lin, and Paul Groth. 2021. Reinforcement Learning–based Collective Entity Alignment with Adaptive Features. *ACM Trans. Inf. Syst.* 39, 3, Article 26 (May 2021), 31 pages.
https://doi.org/10.1145/3446428

X. Zhao and J. Tang were partially supported by Ministry of Science and Technology of China under Grant No. 2020AAA0108802, NSFC under Grants No. 61872446 and No. 71971212, NSF of Hunan Province under Grant No. 2019JJ20024. X. Zhao and J. Tang were also supported by The Science and Technology Innovation Program of Hunan Province under Grant No. 2020RC4046. W. Zeng was partially supported by Postgraduate Scientific Research Innovation Project of Hunan Province under Grant No. CX20190033. X. Lin was partially supported by Grants No. ARC DP200101338, No. ARC DP180103096, and No. ARC DP170101628. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.
Authors' addresses: W. Zeng, X. Zhao (corresponding author), and J. Tang, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, 410072, China; emails: {zengweixin13, xiangzhao, jiuyang_tang}@nudt.edu.cn; X. Lin, The University of New South Wales, Sydney, NSW 2052, Australia; email: lxue@cse.unsw.edu.au; P. Groth, University of Amsterdam, Amsterdam, 1090 GH, The Netherlands; email: p.groth@uva.nl.

1046-8188/2021/05-ART26 $15.00
https://doi.org/10.1145/3446428

1 INTRODUCTION

Knowledge graphs (KGs) play a pivotal role in tasks such as information retrieval [54], question answering [24] and recommendation systems [11]. Although many KGs have been constructed over recent years, none of them can guarantee *full coverage* [37]. Indeed, KGs often contain complementary information, which motivates the task of merging KGs.

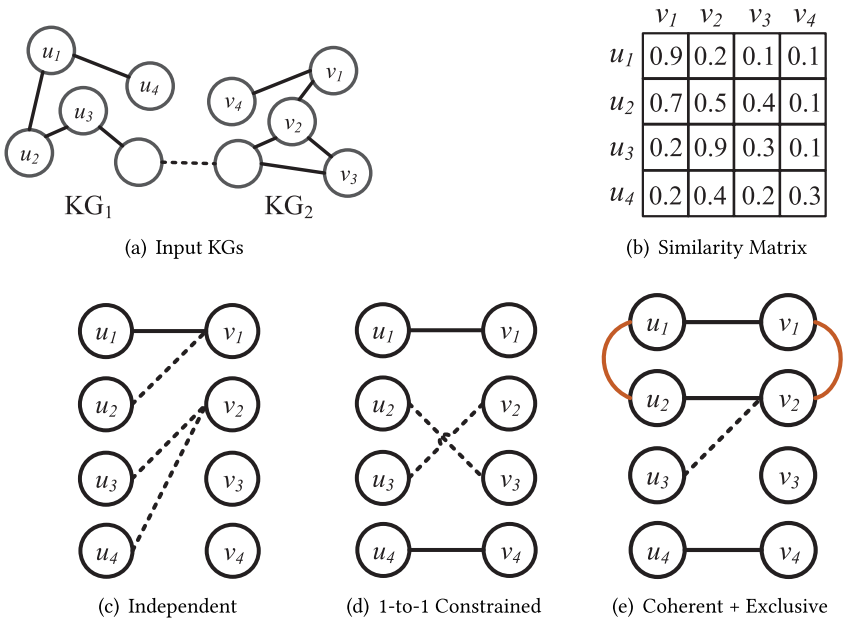
To incorporate the knowledge from a *target* KG into a *source* KG, an important step is to align entities between them. To this end, the task of *entity alignment* (EA) is proposed, which aims to discover entities that have the same meaning but belong to different original KGs. To handle the task, state-of-the-art EA methods [10, 38, 45, 63] assume that equivalent entities in different KGs have similar neighborhood structures. As a consequence, they first use representation learning technologies (e.g., TransE [7] and graph convolutional network (GCN) [25]) to capture structural features of KGs; that is, they map entities into data points in a low-dimensional feature space, where pair-wise similarity can be easily evaluated between the data points. Then, to determine the alignment result, they treat entities *independently* and retrieve a ranked list of target entities for each source entity according to the pair-wise similarities, where the top-ranked target entity is selected as the predicted match to the source entity in question.

Nevertheless, merely using the KG structure, in most cases, cannot guarantee satisfactory alignment results [12]. Consequently, recent methods exploit multiple types of features, e.g., attributes [48, 50], entity description [12, 57], entity names [52, 53], to provide a more comprehensive view for alignment. To fuse different features, these methods first calculate pair-wise similarity scores within each feature-specific space, and then combine these scores to generate the final similarity score. However, they manually assign the weights of features, which can be impractical when the number of features increases, or the importance of certain features varies greatly under different settings.

In response, we first exploit the structural, semantic, string-level features to capture different aspects of the similarity between the entities in source and target KGs. Then, to effectively aggregate different features, we devise an adaptive feature fusion strategy to fuse the feature-specific similarity matrices. The similarity matrices are calculated using the nature-inspired Bray-Curtis dissimilarity [8], a widely used measure to quantify the compositional difference between two coenoses, which can better capture the similarity between entities than commonly used distance measures, e.g., Manhattan distance [50, 52, 53], Euclidean distance [13, 57, 62], and cosine similarity [44–46]. After obtaining the similarity matrix that encodes alignment signals from different features, the next crucial step is to make alignment decisions based on this matrix. As mentioned above, current methods adopt the independent decision-making strategy to generate alignment results, which is illustrated in Example 1.

Example 1. In Figure 1(a) are two KGs (KG_1 and KG_2), where the dashed lines indicate known alignment (i.e., seeds). The target entities v_1, v_2, v_3, and v_4 in KG_2 are to be aligned, respectively, to the source entities u_1, u_2, u_3, and u_4 in KG_1 (i.e., ground truth). Assume that we now arrive at some similarity matrix in Figure 1(b) for the entities in question, where greater values indicate higher similarities. Following the aforementioned independent decision-making paradigm, for the source entity u_1, as the target entity v_1 is of the highest similarity score, (u_1, v_1) is predicted as a match; similarly, (u_2, v_1), (u_3, v_2) and (u_4, v_2) are predicted to be equivalent. Compared with the ground truth, the former is correct, and yet the latter three are erroneous.

Example 1 shows that the independent decision-making strategy fails to produce satisfactory alignment results, as it neglects the interdependence among the decision-making for each source entity. Hence, a few very recent methods [55, 58] incorporate the 1-to-1 constraint to coordinate the alignment process and align entities collectively. This constraint requires that each source entity



	v_1	v_2	v_3	v_4
u_1	0.9	0.2	0.1	0.1
u_2	0.7	0.5	0.4	0.1
u_3	0.2	0.9	0.3	0.1
u_4	0.2	0.4	0.2	0.3

Fig. 1. An example of EA and different strategies. (a) Input KGs, where the nodes (respectively, lines) depict entities (respectively, relations); (b) similarity matrix of all-pair entities; (c)—(e) alignment produced by different strategies, where the entities connected by solid lines are correct matches, whereas those connected by dashed lines are erroneous matches, and the colored lines denote related entities in the same KG.

is matched to exactly one target entity, and vice versa, which has been shown to improve the alignment results. However, we find that the decision-making paradigm incorporating the 1-to-1 constraint might not suffice to produce a satisfying alignment, which is shown in Example 2. This motivates us to find a better characterization of the interdependence. Therefore, we propose a new coordination strategy, which is comprised of the *coherence* and *exclusiveness* constraints, to restrict the collective alignment. Specifically, coherence aims to keep the EA decisions coherent for closely-related entities. Suppose a source entity u is matched with the target entity v; coherence requires that, for the source entities that are closely related to u, their corresponding target entities should also be closely related to the target entity v. Meanwhile, exclusiveness aims to avoid assigning the same target entity to multiple source entities, which requires that, if an entity is already matched, it is less likely to be matched to other entities. In essence, exclusiveness relaxes the 1-to-1 constraint by allowing some extreme cases, e.g., two source entities both have very high confidence to align the same target entity, and produces a pairing that does not necessarily follow the 1-to-1 constraint.

Example 2. Further to Example 1, by imposing the 1-to-1 constraint, one would arrive at the solution that aligns u_3 to v_2, since u_3 has a higher similarity score with v_2 than u_2, and hence u_2 is compelled to match v_3, u_4 is compelled to match v_4. In this case, although u_4 finds the correct match, u_2 and u_3 still deviate from their correct counterparts.

As for our proposed coordination strategy, the exclusiveness constraint would help prevent u_2 from matching v_1, since v_1 has been confidently aligned to u_1; it would also encourage u_4 to align v_4 rather than v_2, since v_2 has higher similarity scores with other entities. The coherence constraint would suggest matching u_2 with v_2, as both u_2 and v_2 are related with the previously matched pair (u_1, v_1). However, the local similarity score between (u_3, v_2) is much higher than (u_2, v_2). In this case, the exclusiveness would allow both u_2 and u_3 to match entity v_2. In comparison with the

cases mentioned above, practicing the new coordination strategy reduces the number of incorrect matches.

To implement the new coordination strategy, we cast the alignment process to the classic sequence decision problem. Given a sequence of source entities, the goal of the sequence decision problem is to decide to which target entity each source entity aligns. We approach the problem with a reinforcement learning (RL)–based framework, which learns to optimize the decision-making for all entities, rather than optimize every single decision separately. More specifically, to implement the coherence and exclusiveness constraints under the RL-based framework, we propose the following design. When making the alignment decision for each source entity, the model takes as input not only local similarity scores but also coherence and exclusiveness signals generated by previously aligned entities. Further, we incorporate the coherence and exclusiveness information into the reward shaping. Thus, the interdependence between EA decisions can be adequately captured.

Contributions. This article is an extended version of our previous work [58]. In this extension, we make substantial improvement:

- We extend the idea of resolving EA jointly by offering an RL-based collective EA framework (CEAFF) that can model both the *coherence* and *exclusiveness* of EA decisions.
- We introduce an adaptive feature fusion strategy to better integrate different features without the need of training data or manual intervention.
- We adopt the Bray-Curtis dissimilarity to measure the similarity between entity embeddings, which applies normalization over the calculated distance and leads to better results compared with the commonly used distance measures such as Manhattan distance and cosine similarity.
- We add some very recent methods for comparison and conduct a more comprehensive analysis.

The main contributions of the article can be summarized as follows:

- We identify the deficiency of existing EA methods in making alignment decisions, and propose a novel solution CEAFF to boost the overall EA performance. This is done by (1) casting the alignment process into the sequence decision problem, and offering a RL-based model to align entities collectively; and (2) exploiting representative features to capture different aspects of the similarity between entities, and integrating them with adaptively assigned features.
- We empirically evaluate our proposal on both cross-lingual and mono-lingual EA tasks against 19 state-of-the-art methods, and the comparative results demonstrate the superiority of CEAFF.

Organization. In Section 2, we formally define the task of EA and introduce related work. In Section 3, we present the outline of CEAFF. In Section 4, we introduce the feature generation process. In Section 5, we introduce the adaptive feature fusion strategy. In Section 6, we elaborate the RL-based collective EA strategy. In Section 7, we introduce the experimental settings. In Section 8, we report and analyze the results. In Section 9, we present the conclusion and future works.

2 PRELIMINARIES

In this section, we first formally define the task of EA, and then introduce the related work.

2.1 Task Definition

The task of EA aims to align entities in different KGs. A KG $G = (E, R, T)$ is a directed graph comprising a set of entities E, relations R, and triples T. A triple $t = (e_i, r_{ij}, e_j) \in T$ represents that a head entity e_i is connected to a tail entity e_j via a relation r_{ij}.

The inputs to EA are a source KG $G_1 = (E_1, R_1, T_1)$, a target KG $G_2 = (E_2, R_2, T_2)$, a set of seed entity pairs $S = \{(u_s, v_s) | u_s \in E_1^s, v_s \in E_2^s, u_s \leftrightarrow v_s\}$, where E_1^s and E_2^s denote the source and target entities in the training set, respectively, $u_s \leftrightarrow v_s$ indicates the source entity u_s and the target entity v_s are *equivalent*, i.e., u_s and v_s refer to the same real-world object. The task of EA is defined as finding a target entity v^* for each source entity u in the test set, i.e., $\Psi = \{(u, v^*) | u \in E_1^t, v^* \in E_2^t\}$, where E_1^t and E_2^t represent the source and target entities in the test set, respectively. A pair of aligned entities $\phi \in \Psi$ is also called a *correspondence*. If the source and target entities in the *correspondence* are *equivalent*, then we call it a *correct correspondence*.

2.2 Related Work

Entity Resolution and Instance Matching. While the problem of EA was introduced a few years ago, the more generic version of the problem—identifying entity records referring to the same real-world entity from different data sources—has been investigated from various angles by different communities [61]. It is mainly referred to as entity resolution (ER) [20, 35], entity matching [16, 34], or record linkage [14]. These tasks assume the inputs are *relational data*, and each data object usually has a large amount of textual information described in multiple attributes. Since EA pursues the same goal as ER, it can be deemed a special but non-trivial case of ER, which aims to handle KGs and deal exclusively with binary relationships, i.e., graph-shaped data [61].

We are aware that there are some collective ER approaches targeted at graph-structured data [4], represented by PARIS [42] and SiGMa [28]. To model the relations among entity records, they adopt collective alignment algorithms such as similarity propagation [1], the LDA model [3], the conditional random fields model [32], Markov logic network models [41], or probabilistic soft logic [26]. These approaches are frequently related to instance matching (or A-Box matching), which aims to find correspondences between entities in different ontologies. Since these methods are established in a setting similar to EA, we include them in the experimental study, and use *collective ER approaches* as the general reference to them. Note that different from these methods, EA solutions build on the recent advances in deep learning and mainly rely on graph representation learning technologies to model the KG structure and generate entity embeddings for alignment [61].

Entity Alignment. A shared pattern can be observed from current EA approaches. First, they generate for each feature a unified embedding space where the entities from different KGs are directly comparable. A frequently used feature is the KG structure, as the equivalent entities in different KGs tend to possess very similar neighboring information. To generate structural entity embeddings, *structure encoders* are devised, including KG representation–based models [12, 13, 44, 45, 48, 62, 63], e.g., TransE [7], and graph neural network (GNN)–based models [10, 30, 50, 52, 56], e.g., GCN [25]. Other available features include attributes [44, 48, 50, 57, 60], entity names [52, 56, 60] and entity descriptions [12, 57]. Correspondingly, *additional information encoders* are designed to embed these features into vector representations. To project the feature-specific embeddings from different KGs into a unified space, they devise unification functions, e.g., the margin-based loss function [10, 30, 50, 52, 53, 57] and the transition function [12, 13, 62], or exploit the corpus fusion strategy [23, 45, 63].

Then, they determine the most likely target entity for each source entity, given the feature-specific unified embeddings. The most common approach is to return a ranked list of target entities

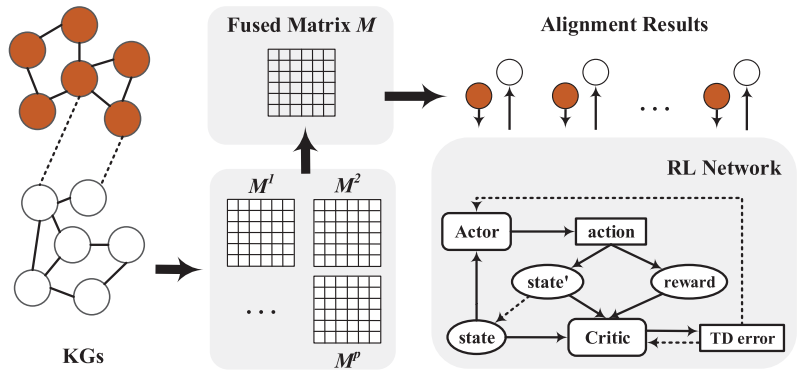


Fig. 2. The framework of CEAFF. The dashed lines in the RL Network represent the *update* operation.

for each source entity according to a specific *distance measure* between embeddings, among which the top ranked entity is regarded as the match. Frequently used distance measures include the Euclidean distance [13, 57, 62], the Manhattan distance [50, 52, 53], and the cosine similarity [44–46].[1] If multiple features are adopted, then it is of necessity to aggregate these features. Some *feature fusion* methods to this end are *representation-level*, which directly combine the feature-specific embeddings and generate an aggregated entity embedding for each entity [52, 53, 56, 60]. Then the similarity between the aggregated entity embeddings is used to characterize the similarity between entities. Other approaches are *outcome-level*, which combine the feature-specific similarity scores with hand-tuned weights [36, 50]. Notably, some very recent works [55, 58] propose to exert the 1-to-1 constraint into the alignment process, which can align entities jointly and generate more accurate results. Specifically, CEA [58] formulates EA as a stable matching problem [21], and uses the deferred acceptance algorithm [39] to produce the results, i.e., no pair of two entities from the opposite side would prefer to be matched to each other rather than their assigned partners [17]. GM-EHD-JEA [55] views EA as a task assignment problem, and employs the Hungarian algorithm [27] to find a set of 1-to-1 alignments that maximize the total pair-wise similarity.

There are some recent studies that improve EA results with the bootstrapping strategy [45, 62] or relation modeling [31, 47, 53].

Reinforcement Learning. Over recent years, reinforcement learning (RL) has been frequently used in many sequence decision problems [18]. Clark and Manning apply RL on coreference resolution to directly optimize a neural mention-ranking model for coreference evaluation metrics, which avoids the need for carefully tuned hyper-parameters [15]. Fang et al. convert entity linking into a sequence decision problem and use an RL model to make decisions from a global prospective [18]. Feng et al. propose an RL-based model for sentence-level relation classification from noisy data, where a selection decision is made for each sentence in the sentence sequence using a policy network [19]. Inspired by these efforts, we also consider EA as the sequence decision process and harness RL to make alignment decisions.

3 THE FRAMEWORK OF OUR PROPOSED MODEL

As shown in Figure 2, there are three stages in our proposed framework:

[1]Note that the distance score between entities can be easily converted to the similarity score by subtracting the distance score from 1. Therefore, in this article, we may use *distance* and *similarity* interchangeably.

- *Feature Generation.* This stage generates representative features for EA, including the structural embeddings learned by GCN, the semantic embeddings represented as averaged word embeddings, and the string similarity matrix calculated using the Levenshtein distance [29].
- *Adaptive Feature Fusion.* This stage fuses various features with adaptively generated weights. We first convert the structural and semantic representations into corresponding similarity matrices using the Bray-Curtis dissimilarity. Then, we combine different features with adaptively assigned weights and generate a fused similarity matrix.
- *Collective EA.* This stage takes the fused similarity matrix as input and generates alignment results by capturing the interdependence between decisions for different entities. Concretely, we convert EA into the sequence decision problem and adopt a deep RL network to model both the coherence and exclusiveness of EA decisions.

Next, we elaborate the modules of CEAFF.

4 FEATURE GENERATION

In this section, we introduce the features employed in CEAFF.

4.1 Structural Information

We harness the graph convolutional network (GCN) [25] to encode the neighborhood information of entities as real-valued vectors.[2] Then, we briefly introduce the configuration of GCN for the EA task, and leave out the GCN fundamentals in the interest of space.

Model Configuration and Training. GCN is harnessed to generate structural representations of entities. We build two two-layer GCNs, and each GCN processes one KG to generate the embeddings of entities. The initial feature matrix, $\boldsymbol{X}$, is sampled from the truncated normal distribution with L2-normalization on rows.[3] It gets updated by the GCN layers, and thus the final output matrix $\boldsymbol{Z}$ can encode the neighboring information of entities. The adjacency matrix $\boldsymbol{A}$ is constructed according to Reference [50]. Note that the dimensionality of the feature vectors is fixed at d_s and kept the same for all layers. The two GCNs share the same weight matrix in each layer.

Then, we project the entity embeddings generated by two GCNs into a unified embedding space using pre-aligned EA pairs S. Specifically, the training objective is to minimize the margin-based ranking loss function:

$$L = \sum_{(u_s, v_s) \in S} \sum_{(u'_s, v'_s) \in S'_{(u_s, v_s)}} [||\boldsymbol{u}_s - \boldsymbol{v}_s||_{l1} - ||\boldsymbol{u}'_s - \boldsymbol{v}'_s||_{l1} + \epsilon]_+, \tag{1}$$

where $[x]_+ = max\{0, x\}$, $S'_{(u_s, v_s)}$ denotes the set of negative EA pairs obtained by corrupting (u_s, v_s), i.e., substituting u_s or v_s with a randomly sampled entity from its corresponding KG. $\boldsymbol{u}_s$ and $\boldsymbol{v}_s$ denotes the (structural) embedding of entity u_s and v_s, respectively. ϵ is a positive margin that separates the positive and negative EA pairs. Stochastic gradient descent is harnessed to minimize the loss function.

4.2 Semantic Information

For each entity, there is textual information associated with it, ranging from its name, its description, to its attribute values. This information can help match entities in different KGs, since

[2]We reckon that there are more advanced frameworks, e.g., the dual-graph convolutional network [52] and the recurrent skipping network [23], for learning the structural representation. They can also be easily plugged into our overall framework after parameter tuning. We will explore these options in the future, as they are not the focus of this article.
[3]Other methods of initialization are also viable. We stick to the random initialization to capture the "pure" structural signal.

equivalent entities share the same meaning. Among this information, the entity name, which identifies an entity, is the most universal textual form. Also, given two entities, comparing their names is the easiest approach to judge whether they are the same. Therefore, in this work, we propose to utilize entity names as the source of textual information.

Entity names can be exploited from the semantic- and string-level. We first introduce the *semantic similarity*. More specifically, we use the averaged word embeddings to capture the semantic meaning of entity names. For a KG, the name embeddings of all entities are denoted in matrix form as $\boldsymbol{N}$. Like word embeddings, similar entity names will be placed adjacently in the entity name representation space.

4.3 String Information

Current methods mainly capture the semantic information of entities. In this work, we contend that, the string information, which has been largely overlooked by current EA literature, is also a contributive feature, since: (1) string similarity is especially useful in tackling the mono-lingual EA task or the cross-lingual EA task where the languages of the KG pair are closely-related (e.g., English and German); and (2) string similarity does not rely on external resources, e.g., pre-trained word embeddings. In particular, we adopt the Levenshtein distance [29], a string metric for measuring the difference between two sequences. We denote the string similarity matrix calculated by the Levenshtein ratio as $\boldsymbol{M^l}$.

5 ADAPTIVE FEATURE FUSION

In this section, we first introduce a new measure to capture the similarity between entity embeddings. Then we point out the limitations of current fusion methods. Finally, we elaborate our proposed adaptive feature fusion strategy.

5.1 Distance Measures

Existing solutions characterize the similarity between entity embeddings by the Manhattan distance [50, 52, 53], the Euclidean distance [13, 57, 62], and the cosine similarity [44–46]. Given two entities u and v with embeddings $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, the Manhattan distance is formalized as

$$D_m(u, v) = \sum_{i=1}^{n} |u_i - v_i|, \tag{2}$$

and the corresponding similarity score is $1 - D_m(u, v)$. Similarly, the Euclidean distance is

$$D_e(u, v) = \sum_{i=1}^{n} \|u_i - v_i\|_2, \tag{3}$$

and the corresponding similarity score is $1 - D_e(u, v)$. Nevertheless, these measures fail to apply normalization over the calculated distance. As a remedy, we use the Bray-Curtis dissimilarity to measure the distance between entity embeddings:

$$D_b(u, v) = \sum_{i=1}^{n} \frac{|u_i - v_i|}{|u_i + v_i|}, \tag{4}$$

and accordingly, the similarity score is $1 - D_b(u, v)$.

Admittedly, the cosine similarity also applies normalization to obtain the similarity score between two embeddings:

$$Sim_c(u, v) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|_2 \, \|\boldsymbol{v}\|_2}. \tag{5}$$

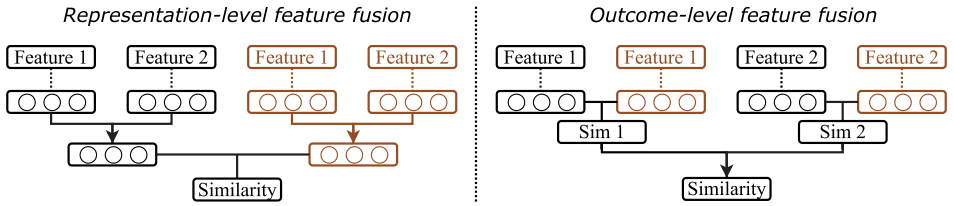


Fig. 3. Different strategies of feature fusion. The operations for different entities are represented in different colors. Dashed lines denote generating vector representations of features. Solid lines without arrows represent calculating the similarity between embeddings, while solid lines with arrows represent the feature fusion process.

However, unlike the Bray-Curtis dissimilarity that performs normalization for *each pair* of elements in the vectors, the cosine similarity directly normalizes the dot product of two vectors. We empirically evaluate these distance measures, and demonstrate the superiority of the Bray-Curtis dissimilarity in Section 8.3.

Given the learned structural embedding matrix $\boldsymbol{Z}$ and the entity name embedding matrix $\boldsymbol{N}$ from Sections 4.1 and 4.2, respectively, we use the Bray-Curtis dissimilarity to generate pair-wise similarity scores between entities. We denote the resultant structural similarity matrix as $\boldsymbol{M^s}$, where rows represent the source entities in the test set, columns denote the target entities in the test set, and each element in the matrix denotes the structural similarity score between a pair of source and target entities. Similarly, the resultant semantic similarity matrix is represented as $\boldsymbol{M^n}$.

5.2 Feature Fusion

Different Strategies of Feature Fusion. To fuse different features, the state of the art directly combines feature-specific embeddings and generates an aggregated embedding for each entity, which is then used to calculate pair-wise similarity between entities (shown in the left of Figure 3). Specifically, some design aggregation strategies to integrate multiple view-specific entity embeddings [60], while some treat certain features as inputs for learning representations of other features [52, 56]. Nevertheless, these *representation-level* fusion techniques might fail to maintain the characteristics of the original features, e.g., two entities might be extremely similar in feature-specific embedding spaces, while placed distantly in the unified representation space.

In this work, we resort to the *outcome-level* feature fusion, which operates on the intermediate outcomes—feature-specific similarity matrices. Current approaches to this end first calculate pair-wise similarity scores within each feature-specific space, and then combine these scores to generate the final similarity score [44, 50, 58] (shown in the right of Figure 3). However, they manually assign the weights of features, which can be inapplicable when the number of features increases, or the importance of certain features varies greatly under different settings. Therefore, it is of significance to dynamically determine the weight of each feature.

Adaptive Feature Fusion Strategy. In this work, we offer an adaptive feature fusion strategy that can dynamically determine the weights of features without the training data. It consists of the following stages:

Confident Correspondence Generation. The inputs to this stage are p features and their corresponding similarity matrices—$\boldsymbol{M^1}, \boldsymbol{M^2}, \ldots, \boldsymbol{M^p}$. $\boldsymbol{M}^p_{ij}$ denotes the similarity between the source entity u_i and the target entity v_j, measured by the feature p. If $\boldsymbol{M}^p_{ij}$ is the largest both along the row and the column, then we consider (u_i, v_j) to be a *confident correspondence* generated by the feature p. This is a relatively strong constraint, and the resulting correspondences are very likely to be correct.

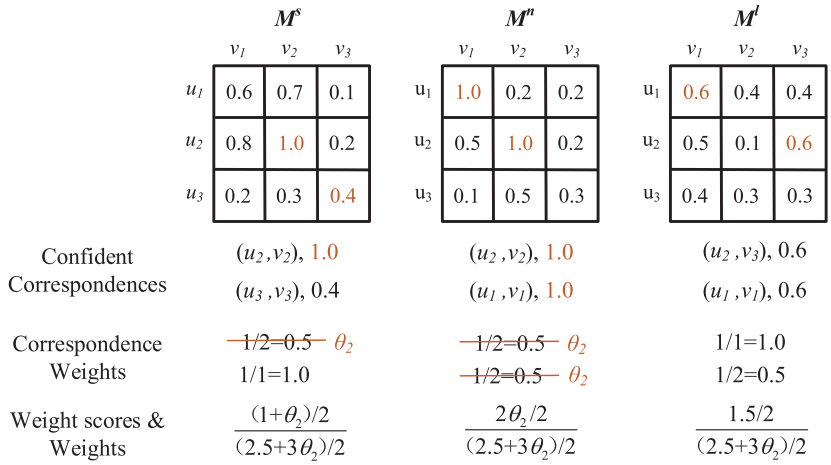


Fig. 4. The framework of adaptive weight assignment.

As thus, we assume that the confident correspondences generated by a specific feature can reflect its importance. The concrete correlation is formulated later.

As shown in Figure 4, there are three feature matrices, $\boldsymbol{M^s}, \boldsymbol{M^n}, \boldsymbol{M^l}$, each of which generates two confident correspondences with similarity scores.

Correspondence Weight Calculation. Then, we determine the weight of each confident correspondence. Instead of assigning equal weights, we assume that the importance of a correspondence is inversely proportional to the number of its occurrences. Specifically, if a correspondence is generated by q features, then its weight is set to $1/q$, as it is believed that the frequently occurring correspondence brings less new information in comparison with a correspondence that has the quality of being detected by only one single feature matrix [22].

Additionally, for a correspondence with very large similarity score $\boldsymbol{M}^p_{ij} > \theta_1$, we set its weight to a small value θ_2. With this setting, the features that are very effective would not be assigned with extremely large weights, and the less effective features can still contribute to the alignment. We empirically validate the usefulness of this strategy in Section 8.4.

As shown in Figure 4, regarding the two correspondences generated by $\boldsymbol{M^s}$, (u_3, v_3) is assigned with weight $1/1 = 1$, as it is only produced by $\boldsymbol{M^s}$, while (u_2, v_2) is detected by both $\boldsymbol{M^s}$ and $\boldsymbol{M^n}$, and hence endowed with the weight $1/2 = 0.5$. Moreover, the weight of (u_2, v_2) is reset to θ_2, since its similarity score exceeds $\theta_1 = 0.95$.

Feature Weight Calculation. After obtaining the weight of each confident correspondence, the *weight score* of feature p is defined as the sum of the weights of its confident correspondences, divided by the number of these correspondences. The *weight* of feature p is the ratio between its weight score and the total weight score of all features.

Feature Fusion with Adaptive Weight. We combine the feature-specific similarity matrices with their adaptively assigned weights to generate the fused similarity matrix $\boldsymbol{M}$. Particularly, in this work, we utilize three representative features—structural-, semantic-, and string-level features, which are elaborated in Section 4.

Remark. One possible solution to determine the weight of each feature is using machine learning techniques to learn the weights. Nevertheless, for each source entity, the number of negative target entities is much larger than the positive ones. Besides, the amount of training data is very limited.

Such restraints make it difficult to generate high-quality training corpus, and hence might hamper learning appropriate weights. We will report the performance of the learning-based approach in Section 8.4.

6 RL-BASED COLLECTIVE EA STRATEGY

After obtaining the fused similarity matrix $\boldsymbol{M}$, we proceed to the alignment decision-making stage, which is the core step of the EA task and can be regarded as a matching process, i.e., matching a source entity with a target entity. The majority of existing works choose the optimal local match for each source entity while neglecting the influence between matching decisions. To capture the interdependence among alignment decisions, CEA [58] and GM-EHD-JEA [55] exert the 1-to-1 constraint on the matching process. More specifically, CEA adopts the deferred acceptance algorithm to find a stable matching result, which has the worst runtime complexity of $O(n^2)$. The resulting matching is locally optimal, i.e., a man-optimal assignment (the man refers to the source entity in our case). Meanwhile, GM-EHD-JEA frames the matching process as the task assignment problem (maximizing the local similarity scores under the 1-to-1 constraint), which is essentially a fundamental combinatorial optimization problem whose exact solution can be found by the Hungarian algorithm. The Hungarian algorithm has the worst runtime complexity of $O(n^3)$, and GM-EHD-JEA employs a search space separation strategy to reduce the time cost (which also hurts the performance).

There are several main drawbacks of framing the alignment process as the matching problem: (1) High worst time complexity: $O(n^2)$ for the deferred acceptance algorithm and $O(n^3)$ for the Hungarian algorithm; and (2) Difficulty of including other constraints. As stated in the introduction, the coherence constraint is also of importance. Nevertheless, it is hard to incorporate such a constraint into the stable matching process. For the combinatorial optimization problem, although we can consider the coherence and define the goal as maximizing the local similarity scores *and* the overall coherence, the resulting optimization problem becomes NP-hard (similar to the global optimization in Entity Linking task [40]). Besides, there is no straightforward approach to replace the 1-to-1 constraint in the matching process with the exclusiveness constraint.

6.1 EA as a Sequence Decision Problem

In this work, we cast EA to the classic sequence decision problem; that is, given a sequence of source entities, EA generates for each source entity a target entity. This problem can be solved by a RL-based model. In this model, source entities are aligned in a sequential but collective manner, which allows every decision to be made based on current state and previous ones, such that the interdependence among the decisions is captured. Particularly, we characterize the interdependence by a novel coordination strategy, which comprises the *exclusiveness* constraint—a previously chosen target entity is less likely to be assigned to the rest of the source entities, and the *coherence* constraint—the relevance between entities can assist the decision-making.

As for the RL framework, we adopt the Advantage Actor-Critic (A2C) [33] model, where the Actor conducts actions in an environment and the Critic computes the value functions to help the Actor in learning. These two agents participate in a game where they both get better in their own roles as the learning gets further. As shown in Figure 2, at each step, the RL network takes the current state as input, outputs an action (a target entity), which influences the following state and generates a reward. Taking the current state, the next state, and the reward as input, the Critic calculates the Temporal-Difference (TD) error and updates its network. The TD error is then fed to the Actor for the parameter updates.

Next, we introduce the basic components of the RL model, as well as the optimization and learning process.

6.2 RL-based Collective EA Model

Environment. The environment includes the source and target entities in the test set, the adjacency matrices of KGs, as well as the fused similarity matrix $\boldsymbol{M}$.

State. The state vector $\boldsymbol{s}$ is expressed as $\boldsymbol{s^1} \circ \boldsymbol{s^2} + \boldsymbol{s^3}$, where $\circ$ represents element-wise product, and $\boldsymbol{s^1}$, $\boldsymbol{s^2}$, $\boldsymbol{s^3}$ refer to the *local similarity vector*, the *exclusiveness vector* and the *coherence vector*, respectively. More specifically,

- The *local similarity vector* is obtained from the fused similarity matrix $\boldsymbol{M}$, which characterizes the similarity between the current source entity and the candidate target entities.
- The *exclusiveness vector* indicates the target entities that have been chosen. Each element in the vector corresponds to a target entity, and the value can be chosen between "1" and "−1." While "1" denotes that the corresponding target entity has not been chosen yet, "−1" denotes that the corresponding target entity has been chosen. The exclusiveness vector is initialized with "1"s; then if a target entity is chosen, the value in the corresponding position is replaced with "−1."
- The *coherence vector* characterizes the relevance between current candidate target entities and previously chosen target entities. Concretely, given the current source entity, we first retrieve its related source entities that have been matched according to the adjacency matrix in the source KG. We consider the target entities chosen by these source entities as the *contextual entities* for aligning the current source entity. For each candidate target entity, we define its coherence score as the number of *contextual entities* it is directly connected with in the KG. The coherence scores of all candidate target entities constitute the *coherence vector*.

Action. An action denotes the Actor choosing a target entity for a source entity given a particular state. The Actor takes the current state as input, and chooses a target entity from the candidate target entities according to the probabilities calculated by a neural model. The neural network (parameterized by θ) consists of a fully connected layer and a softmax layer. More specifically, the input state at step i is fed into the fully connected layer, which generates a hidden state:

$$\boldsymbol{h}(s_i) = Relu(\boldsymbol{W_1 s_i} + \boldsymbol{b_1}), \tag{6}$$

where $\boldsymbol{W_1}$ and $\boldsymbol{b_1}$ are the parameters of this layer. The hidden state is then fed to the softmax layer, which generates the probability distribution of actions:

$$\pi_\theta(a|s_i) = Softmax(\boldsymbol{W_2 h}(s_i) + \boldsymbol{b_2}), \tag{7}$$

where $\boldsymbol{W_2}$ and $\boldsymbol{b_2}$ are the parameters of the softmax layer. Then, the target entity is sampled from the candidate target entities, according to $\pi_\theta(a|s_i)$. Notably, to reduce the model complexity and increase the efficiency, for each source entity, we merely consider the top-τ ranked target entities as the candidate entities according to the fused similarity scores.

Reward. The reward is a feedback indicating the influence caused by the action. Particularly, we want to stimulate the action that maximizes the local similarity and global coherence, and discourage the action that has been taken previously. As thus, we define the reward at step i as $r_{i+1} = \boldsymbol{s^1}(a_i) \cdot \boldsymbol{s^2}(a_i) + \boldsymbol{s^3}(a_i)$, where a_i represents the chosen action at step i.

With these components, a viable solution is to apply the REINFORCE algorithm [51] with policy gradients to maximize the total reward and determine the parameters of the neural network. Nevertheless, the drawback of this approach is that the reward can only be calculated after the whole episode is finished, while the situation in each step cannot be well characterized. To address this

issue, we design a Critic model that approximates the value function of each state and makes an update at each step.

Critic Network. The Critic network comprises two fully connected layers (parameterized by η). At step i, the first layer takes as input the state vector, and outputs a hidden vector:

$$\boldsymbol{h_c}(s_i) = Relu(\boldsymbol{W_3 s_i} + \boldsymbol{b_3}). \tag{8}$$

Then, the next layer converts the hidden vector into an estimated value:

$$V_\eta(s_i) = \boldsymbol{W_4 h_c}(s_i) + \boldsymbol{b_4}, \tag{9}$$

where $\boldsymbol{W_3}, \boldsymbol{W_4}, \boldsymbol{b_3}, \boldsymbol{b_4}$ are the learnable parameters.

Optimization and Learning Procedure. The training of the Actor and Critic networks are performed separately. Regarding the Actor, the objective is to obtain the highest total reward, which can be formally defined as

$$J(\theta) = \mathbb{E}\left[\sum_{i=0}^{n-1} r_{i+1}|\pi_\theta\right] = \sum_{i=0}^{n-1} P(s_i, a_i) r_{i+1}, \tag{10}$$

where $P(s_i, a_i)$ denotes the probability of (s_i, a_i), i.e., choosing action a_i for the state s_i. To optimize the objective, following [33], the update of parameter θ can be written as

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(a_i|s_i)(r_{i+1} + \gamma V_\eta(s_{i+1}) - V_\eta(s_i)), \tag{11}$$

where $V_\eta(s_i)$, $V_\eta(s_{i+1})$ correspond to the estimated values generated by the Critic network, γ denotes the decay factor, and $r_{i+1} + \gamma V_\eta(s_{i+1}) - V_\eta(s_i)$ is the Temporal-Difference (TD) error, which can be regarded as a good estimator of the reward at each step. α denotes the learning rate.

As for the Critic, we aim to minimize the mean squared TD error. To achieve the objective, the update of parameter η can be written as

$$\Delta\eta = \beta \nabla_\eta V_\eta(s_i)(r_{i+1} + \gamma V_\eta(s_{i+1}) - V_\eta(s_i)), \tag{12}$$

where β denotes the learning rate.

We illustrate the whole learning procedure in Algorithm 1 and Figure 2. The RL network takes the current state s_u as input, outputs an action a_u, which in turn affects the following state $s_{u'}$ and generates a reward r_u. Taking the current state s_u, the next state $s_{u'}$, and the reward r_u as input, the Critic calculates the TD error and updates its network. The TD error is then fed to the Actor for the network update. This procedure continues until it reaches the final state. Notably, the learning sequence is determined according to the highest local similarity score of each source entity. The source entities with higher maximum similarity scores are dealt first.

Preliminary Treatment. Due to the large number of entities to be aligned, we propose to filter out the source entities that have a high probability of being correctly aligned by merely using the fused similarity matrix $\boldsymbol{M}$. Concretely, for each source entity, if its top-ranked target entity also considers it as the top-ranked source entity, then we assume that this entity pair is highly likely to be correct. Then, we use the rest of the source/target entities as the input to the RL process.

Remark. Different from the current collective strategies, which *explicitly* exert the 1-to-1 constraint into the alignment process, the RL framework *implicitly* models the exclusiveness as a component of the state and the reward. In this way, although the Actor learns to avoid selecting the already chosen target entities, it still takes into account the other aspects of the current state, i.e., the coherence and local similarity, when making the alignment decision. Consequently, the resultant pairing does not strictly follow the 1-to-1 constraint.

ALGORITHM 1: The learning procedure of the A2C network.

Input: E_t^1:source entities; E_t^2: target entities; Adjacency matrices; $\boldsymbol{M}$: similarity matrix
Output: A target entity for each source entity

```
Initialize the network parameters;
foreach epoch do
    foreach u ∈ E_t^1 do
        Generate the state vector s_u, forward it to the Actor;
        Sample an action a_u from π_θ;
        Take the action a_u (target entity v_u), get the reward r_u and next state s_u', forward them to
        Critic;
        Compute the TD error δ = r_u + γV_η(s_u') − V_η(s_u);
        Update the Critic, η ← η + βδ∇_η V_η(s_u);
        Update the Actor, θ ← θ + αδ∇_θ log π_θ(a_u|s_u);
```

7 EXPERIMENTAL SETTING

In this section, we describe the experimental settings, including the research questions (Section 7.1), datasets (Section 7.2), parameter settings (Section 7.3), evaluation metrics (Section 7.4), and methods to compare (Section 7.5). The code of CEAFF and the dataset splits can be accessed via https://github.com/DexterZeng/CEAFF.

7.1 Research Questions

We attempt to answer the following research questions:

(RQ1) Can CEAFF outperform the state-of-the-art approaches on the EA task? (Section 8.1)
(RQ2) Are the features useful? (Section 8.2)
(RQ3) Is the Bray-Curtis dissimilarity a better distance measure than the commonly used measures, i.e., cosine similarity, Manhattan distance and Euclidean distance? (Section 8.3)
(RQ4) Can the adaptive feature fusion strategy effectively integrate features? (Section 8.4)
(RQ5) Does the RL-based collective alignment strategy contribute to the performance of CEAFF? (Section 8.5)
(RQ6) In which cases does CEAFF fail? and why? (Section 8.6)

7.2 Datasets

Three datasets, including eight KG pairs, are used for evaluation:

DBP15K. Sun et al. [44] established the DBP15K dataset. They extracted 15 thousand inter-language links (ILLs) in DBpedia [2] with popular entities from English to Chinese, Japanese and French, respectively. These ILLs are also considered as gold standards.

SRPRS. Guo et al. [23] pointed out that KGs in DBP15K are too dense and the degree distributions of entities deviate from real-life KGs. Therefore, they established a new EA benchmark that follows real-life distribution by using ILLs in DBpedia and reference links among DBpedia, YAGO [43] and Wikidata [49]. They first divided the entities in a KG into several groups by their degrees, and then separately performed random PageRank sampling for each group. To guarantee the distributions of the sampled datasets following the original KGs, they used the Kolmogorov-Smirnov (K-S) test to control the difference. The final evaluation benchmark consists of both cross-lingual and mono-lingual datasets.

Table 1. The Statistics of the Evaluation Benchmarks

Dataset	KG pairs	#Triples	#Entities	#Relations	#Align.	#Test	#Val	#Train
$\text{DBP15K}_{\text{ZH-EN}}$	DBpedia(Chinese)	70,414	19,388	1,701	15,000	10,500	900	3,600
	DBpedia(English)	95,142	19,572	1,323				
$\text{DBP15K}_{\text{JA-EN}}$	DBpedia(Japanese)	77,214	19,814	1,299	15,000	10,500	900	3,600
	DBpedia(English)	93,484	19,780	1,153				
$\text{DBP15K}_{\text{FR-EN}}$	DBpedia(French)	105,998	19,661	903	15,000	10,500	900	3,600
	DBpedia(English)	115,722	19,993	1,208				
$\text{SRPRS}_{\text{EN-FR}}$	DBpedia(English)	36,508	15,000	221	15,000	10,500	900	3,600
	DBpedia(French)	33,532	15,000	177				
$\text{SRPRS}_{\text{EN-DE}}$	DBpedia(English)	38,281	15,000	222	15,000	10,500	900	3,600
	DBpedia(German)	37,069	15,000	120				
$\text{SRPRS}_{\text{DBP-WD}}$	DBpedia	38,421	15,000	253	15,000	10,500	900	3,600
	Wikidata	40,159	15,000	144				
$\text{SRPRS}_{\text{DBP-YG}}$	DBpedia	33,571	15,000	223	15,000	10,500	900	3,600
	YAGO3	34,660	15,000	30				
DBP-FB	DBpedia	96,414	29,861	407	25,542	17,880	1,532	6,130
	Freebase	111,974	25,542	882				

The #Triples, #Entities, and #Relations columns indicate the number of triples, entities, and relations in each KG, respectively. The #Align column indicates the total number of gold entity pairs (reference links). The #Test, #Val, and #Train columns indicate the number of testing, validation, and training entity pairs, respectively.

DBP-FB. Zhao et al. [61] noticed that in existing mono-lingual datasets, equivalent entities in different KGs possess identical names from the entity identifiers, which means that a simple comparison of these names can achieve reasonably accurate results. Therefore, they established a new dataset DBP-FB (extracted from DBpedia and Freebase [6]) to mirror the real-life difficulties.

A concise summary of the statistics of the datasets can be found in Table 1. Note that previous works do not set aside part of the gold standards as the validation set. They use 70% of the gold data as the test set and 30% as the training set. To follow a more standard evaluation paradigm, in this work, we use 20% of the original training set as the validation set for hyper-parameter tuning and used the rest for training. The statistics of the new dataset splits are reported in Table 1. The experiments in this work are all conducted on the new dataset splits.

7.3 Parameter Settings

For learning the *structural representation*, as it is not the focus of this article, we use the popular parameter settings in existing works [50, 52, 58] and set d_s to 300, ϵ to 3, the number of training epochs to 300. Five negative examples are generated for each positive pair. Regarding the *entity name representation*, we utilize the pre-trained fastText word embeddings with subword information [5] as word embeddings and obtain the multilingual word embeddings from MUSE[4]. As for the *adaptive feature fusion*, we set θ_1 to 0.99, θ_2 to 0.48, which are tuned on the validation set. Regarding the *deep RL model*, we set γ to 0.9, α to 0.001, β to 0.01, τ to 10, the dimensionality of the hidden state in Equations (6) to 10, the dimensionality of the hidden state in Equations (8) to 10. We perform two rounds of the preliminary treatment, which will be further discussed in Section 8.5. These hyper-parameters are tuned on the validation set.

[4]https://github.com/facebookresearch/MUSE.

7.4 Evaluation Metrics

We use precision, recall and F1 score as the evaluation metrics. Precision is computed as the number of correct matches produced by a method divided by the number of matches produced by a method. Recall is computed as the number of correct matches produced by a method divided by the total number of correct matches (i.e., gold matches). F1 score is computed as the harmonic mean between precision and recall. Note that in existing EA datasets, all of the entities in a KG are matchable (i.e., for each entity, there exists an equivalent entity in the other KG), and thus the total number of correct matches equals to the number of entities in a KG. Besides, state-of-the-art EA methods [31, 47] generate matches for all the entities in a KG. Therefore, the number of matches produced by these methods is equal to the number of gold matches, and the values of precision, recall, and F1 score are equal for all current EA methods. In the following, unless otherwise specified, we only report the values of precision for these EA methods.

We notice that previous EA methods [13, 50] use Hits@k (k = 1, 10) and mean reciprocal rank (MRR) as the evaluation metrics. For each source entity, they rank the target entities according to the similarity scores in a descending order. Hits@k is computed as the number of source entities whose equivalent target entity is ranked in the top-k results divided by the total number of source entities, while MRR characterizes the rank of the ground truth. Particularly, Hits@1 is the fraction of correctly aligned source entities among all the aligned source entities, and for state-of-the-art EA methods, the Hits@1 value is equal to the precision value, since these methods generate matches for all source entities.

7.5 Methods to Compare

The following state-of-the-art EA methods are used for comparison, which can be divided into two groups—methods merely using structural information, and methods using multi-type information. Specifically, the first group consists of:

- MTransE (2017) [13]: This work uses TransE to learning entity embeddings.
- RSNs (2019) [23]: This work integrates recurrent neural networks (RNNs) with residual learning to capture the long-term relational dependencies within and between KGs.
- MuGNN (2019) [10]: A novel multi-channel graph neural network model is put forward to learn alignment-oriented KG embeddings by robustly encoding two KGs via multiple channels.
- AliNet (2020) [47]: This work proposes an EA network, which aggregates both direct and distant neighborhood information via attention and gating mechanism.
- KECG (2019) [30]: This article proposes to jointly learn knowledge embedding model that encodes inner-graph relationships, and cross-graph model that enhances entity embeddings with their neighbors' information.
- ITransE (2017) [62]: An iterative training process is used to improve the alignment results.
- BootEA (2018) [45]: This work devises an alignment-oriented KG embedding framework and a bootstrapping strategy.
- NAEA (2019) [63]: This work proposes a neighborhood-aware attentional representation method to learn neighbor-level representation.
- TransEdge (2019) [46]: A novel edge-centric embedding model is put forward for EA, which contextualizes relation representations in terms of specific head-tail entity pairs.
- MRAEA (2020) [31]: This solution directly models cross-lingual entity embeddings by attending over the node's incoming and outgoing neighbors and its connected relations' meta semantics. A bi-directional iterative strategy is used to add newly aligned seeds during training.

The second group includes:

- GCN-Align (2018) [50]: This work utilizes GCN to generate entity embeddings and combines them with attribute embeddings to align entities in different KGs.
- JAPE (2017) [44]: In this work, the attributes of entities are harnessed to refine the structural information for alignment.
- RDGCN (2019) [52]: A relation-aware dual-graph convolutional network is proposed to incorporate relation information via attentive interactions between KG and its dual relation counterpart.
- HGCN (2019) [53]: This work proposes to jointly learn entity and relation representations for EA.
- GM-Align (2019) [56]: A local sub-graph of an entity is constructed to represent the entity. Entity name information is harnessed for initializing the overall framework.
- GM-EHD-JEA (2020) [55]: This work introduces two coordinated reasoning methods to solve the many-to-one problem during the decoding process of EA.
- HMAN (2019) [57]: This work combines multi-aspect information to learn entity embeddings.
- CEA (2020) [58]: This work proposes a collective framework that formulates EA as the classic stable matching problem, and solves it with the deferred acceptance algorithm.
- DAT (2020) [59]: This work conceives a degree-aware co-attention network to effectively fuse available features, to improve the performance of entities in tail.

We obtain the results of these methods on the new dataset splits by using the source codes and parameter settings provided by the authors. However, the source codes of GM-EHD-JEA are not available, and we adopt its results on DBP15K from its original paper, where 30% of the gold standards are used for training (which equals to the combination of training and validation sets in this work).[5]

We include a string-based heuristic Lev for comparison, which uses Levenshtein distance to measure the distance between entity names and selects for each source entity the closest target entity as the alignment result.

We also report the performance of some variants of CEAFF:

- CEAFF-SM: Built on the basic components, this approach replaces the RL-based collective strategy with 1-to-1 constrained stable matching.
- CEAFF-Coh: This approach merely models the *coherence* during the RL process.
- CEAFF-Excl: This approach merely models the *exclusiveness* during the RL process.

The best results on each dataset are denoted in **bold**. AVG represents the averaged results. We use the two-tailed t-test to measure the statistical significance. Significant improvements over CEA are marked with $^{\blacktriangle}$ ($\alpha = 0.05$), significant improvements over GM-EHD-JEA are marked with $^{\triangle}$ ($\alpha = 0.05$), and significant improvements over CEAFF-SM are marked with $^{\blacklozenge}$ ($\alpha = 0.05$).

8 RESULTS AND ANALYSES

This section reports the experimental results. We first address RQ1 by comparing CEAFF with state-of-the-art EA and ER solutions on the EA datasets (Section 8.1). Then, by conducting the ablation study and detailed analysis of the components in CEAFF, we answer RQ2, RQ3, RQ4, and

[5]Theoretically, these results should be higher than its actual results on the new splits (where there are less training data) [31].

RQ5 in Sections 8.2, 8.3, 8.4, and 8.5, respectively. Finally, we perform detailed case study and error analysis to answer RQ6 (Section 8.6).

8.1 Performance Comparison

In this subsection, we answer RQ1. We first report and discuss the experimental results of state-of-the-art EA solutions.

Results in the First Group. Solutions in the first group use the structural information for alignment. They can be further divided into two categories depending on the use of iterative training strategy. The first category includes methods that do not employ the bootstrapping strategy, i.e., MTransE, RSNs, MuGNN, AliNet, and KECG. MTransE obtains unsatisfactory results as it learns the embeddings in different vector spaces, and suffers from information loss when modeling the transition between different embedding spaces [52]. RSNs enhances the performance by taking into account the long-term relational dependencies between entities, which can capture more structural signals for alignment. On DBP15K, MuGNN, AliNet, and KECG achieve much better results than MTransE, since they exploit more neighboring information for alignment. Concretely, MuGNN uses a multi-channel graph neural network that captures different levels of structural information. KECG adopts a similar idea by jointly learning entity embeddings that encode both inner-graph relationships and neighboring information. AliNet aggregates both direct and distant neighborhood information via attention and gating mechanism. However, they are still outperformed by RSNs. Additionally, MuGNN attains worse performance than MTransE on SRPRS, since there are no aligned relations on SRPRS, where the rule transferring of MuGNN fails to work.

Methods in the second category employ bootstrapping strategy to enhance the alignment performance, including ITransE, BootEA, NAEA, TransEdge, and MRAEA. BootEA achieves better performance than ITransE, since it devises an alignment-oriented KG embedding framework with one-to-one constrained bootstrapping strategy. On top of BootEA, NAEA uses a neighborhood-aware attentional representation model to make better use of the KG structure and learn more comprehensive structural representations. Nevertheless, using the codes provided by the authors cannot reproduce the results reported in the original paper. TransEdge attains promising results, as it employs an edge-centric embedding model to capture structural information, which generates more precise entity embeddings and hence better alignment results. Similarly, MRAEA proposes to improve EA performance by modeling relation semantics and employing the bi-directional iterative training strategy, and it attains the best results among the methods that merely use the structural information.

Noteworthily, the overall results on SRPRS and DBP-FB are worse than DBP15K, as the KGs in DBP15K are much denser than those in SRPRS and DBP-FB [23, 61].

Results in the Second Group. Taking advantage of the attribute information, GCN-Align, JAPE and HMAN outperform MTransE. HMAN achieves better performance than JAPE and GCN-Align, since (1) it explicitly regards the relation type features as model input; and (2) it employs feedforward neural networks to obtain the embeddings of relations and attributes [57].

The other methods all exploit the entity name information for alignment, and their results exceed the attribute-enhanced methods. This verifies the significance of entity names for aligning entities. Among them, the performance of RDGCN and HGCN are close, surpassing GM-Align. This is because they employ relations to improve entity embeddings, which has been largely neglected in previous GNN-based EA models. DAT also attains promising results by improving the alignment performance of long-tail entities. Both CEA and GM-EHD-JEA focus on the collective alignment process. While GM-EHD-JEA views EA as the task assignment problem and employs

Table 2. The Precision Results on DBP15K and DBP-FB

	DBP15K				DBP-FB
	ZH-EN	JA-EN	FR-EN	AVG	
MTransE	0.172	0.194	0.185	0.184	0.059
RSNs	0.519	0.520	0.565	0.535	0.241
MuGNN	0.361	0.306	0.355	0.341	0.202
AliNet	0.411	0.440	0.433	0.428	0.180
KECG	0.439	0.455	0.453	0.449	0.210
ITransE	0.149	0.197	0.169	0.172	0.018
BootEA	0.574	0.555	0.592	0.574	0.223
NAEA	0.303	0.282	0.297	0.294	/
TransEdge	0.580	0.542	0.573	0.565	0.253
MRAEA	0.617	0.629	0.636	0.627	0.294
GCN-Align	0.401	0.390	0.375	0.389	0.178
JAPE	0.368	0.327	0.273	0.323	0.065
HMAN	0.556	0.555	0.542	0.551	0.274
RDGCN	0.692	0.751	0.876	0.773	0.660
HGCN	0.707	0.746	0.871	0.775	0.789
GM-Align	0.552	0.611	0.768	0.644	0.702
GM-EHD-JEA	0.736	0.792	0.924	0.817	/
CEA	0.776$^{\triangle}$	0.853$^{\triangle}$	0.967$^{\triangle}$	0.865$^{\triangle}$	0.960
Lev	0.070	0.066	0.781	0.306	0.578
CEAFF-SM	0.806$^{\triangle\blacktriangle}$	0.866$^{\triangle\blacktriangle}$	0.971$^{\triangle\blacktriangle}$	0.881$^{\triangle\blacktriangle}$	0.962$^{\blacktriangle}$
CEAFF-Excl	0.789$^{\triangle\blacktriangle}$	0.857$^{\triangle\blacktriangle}$	0.967$^{\triangle}$	0.871$^{\triangle\blacktriangle}$	0.957
CEAFF-Coh	0.807$^{\triangle\blacktriangle}$	0.864$^{\triangle\blacktriangle}$	0.969$^{\triangle\blacktriangle}$	0.880$^{\triangle\blacktriangle}$	0.955
CEAFF	**0.811**$^{\triangle\blacktriangle\blacklozenge}$	**0.868**$^{\triangle\blacktriangle\blacklozenge}$	**0.972**$^{\triangle\blacktriangle\blacklozenge}$	**0.884**$^{\triangle\blacktriangle\blacklozenge}$	**0.963**$^{\blacktriangle\blacklozenge}$

[1]We fail to obtain the results of NAEA on DBP-FB under our experimental environment, as it requires extremely large amount of memory space. We also fail to obtain the result of GM-EHD-JEA on DBP-FB, since its source code is not available and our implementation cannot reproduce the claimed performance.

the Hungarian algorithm, CEA outperforms GM-EHD-JEA by modeling EA as the stable matching problem and solving it with the deferred acceptance algorithm.

Notably, it can be observed from Table 2 that the methods using entity names achieve much better results on the datasets with closely-related language pairs (e.g., FR-EN) than those with distantly related language pairs (e.g., ZH-EN). This shows that the language pairs can influence the use of entity name information and in turn affect the overall alignment performance.

Results of Our Proposal. As can be observed from Tables 2 and 3, our proposal consistently outperforms all other methods on all datasets. We attribute the superiority of our model to its four advantages: (1) we leverage three representative sources of information, i.e., structural, semantic and string-level features, to offer more comprehensive signals for EA; and (2) we adopt the Bray-Curtis dissimilarity to better measure the similarity between entity embeddings; and (3) we fuse the features with adaptively assigned weights, which can fully take into consideration the strength of each feature; and (4) we align the source entities collectively via reinforcement learning, which can adequately capture the interdependence between EA decisions. Particularly, CEAFF achieves better results than the collective alignment methods CEA and GM-EHD-JEA, and the improvements are statistically significant.

Table 3. The Precision Results on SRPRS

	$SRPRS_{EN-FR}$	$SRPRS_{EN-DE}$	$SRPRS_{DBP-WD}$	$SRPRS_{DBP-YG}$	AVG
MTransE	0.166	0.125	0.186	0.158	0.159
RSNs	0.315	0.455	0.380	0.344	0.374
MuGNN	0.114	0.225	0.130	0.159	0.157
AliNet	0.228	0.352	0.258	0.273	0.278
KECG	0.277	0.414	0.306	0.333	0.333
ITransE	0.084	0.088	0.069	0.064	0.076
BootEA	0.351	0.493	0.387	0.379	0.403
NAEA	0.178	0.306	0.187	0.200	0.218
TransEdge	0.355	0.501	0.388	0.379	0.406
MRAEA	0.380	0.531	0.428	0.450	0.447
GCN-Align	0.264	0.395	0.299	0.331	0.322
JAPE	0.236	0.263	0.217	0.189	0.226
HMAN	0.400	0.532	0.431	0.446	0.452
RDGCN	0.580	0.676	0.989	0.993	0.810
HGCN	0.563	0.634	0.988	0.988	0.793
GM-Align	0.570	0.678	0.799	0.764	0.703
DAT	0.765	0.862	0.921	0.931	0.870
CEA	0.966	0.977	1.000	1.000	0.986
Lev	0.851	0.862	**1.000**	**1.000**	0.928
CEAFF-SM	0.966	0.979▲	**1.000**	**1.000**	0.986
CEAFF-Excl	0.964	0.979▲	**1.000**	**1.000**	0.986
CEAFF-Coh	0.964	0.979▲	**1.000**	**1.000**	0.986
CEAFF	**0.968**▲♦	**0.981**▲♦	**1.000**	**1.000**	**0.987**▲♦

Moreover, CEAFF achieves superior results than CEAFF-SM on most datasets, as it simultaneously models the *exclusiveness* and *coherence* of EA decisions. This is also validated by the fact that CEAFF outperforms CEAFF-Excl and CEAFF-Coh, which merely models the *exclusiveness* and *coherence* during the RL process, respectively. Notably, CEAFF advances the precision to 1 on $SRPRS_{DBP-WD}$ and $SRPRS_{DBP-YG}$. This is because the entity names in DBpedia, YAGO and Wikidata are nearly identical, where the string-level feature is extremely effective. In fact, merely using the Levenshtein distance between entity names (Lev) can already attain ground-truth results on these two datasets. In comparison, on DBP-FB, Lev merely attains the precision at 0.578. CEAFF further improves the performance by incorporating the structural and semantic information for alignment.

Evaluation as the Ranking Problem. For the comprehensiveness of the experiment, following previous works, we consider the EA results in the form of ranked target entity lists, and report the Hits@1, 10 and MRR values in Table 4. Note that for the collective EA strategies, i.e., CEA, CEAFF and GM-EHD-JEA, they directly generate the matches rather than the ranked entity lists, and cannot be evaluated by the metrics of the ranking problem, i.e., Hits@1, Hits@10 and MRR. Therefore, we do not include them in Table 4. Instead, we report the results of CEAFF w/o C where the collective alignment component is removed. We leave out the evaluation performance on SRPRS and DBP-FB in the interest of space. It reads from Table 4 that, CEAFF w/o C attains the best overall results.

Comparison with the Collective ER Approach. As mentioned in Section 2, some collective ER approaches design algorithms to handle graph-structured data. Hence, we compare with a

Table 4. Evaluation as the Ranking Problem on DBP15K

	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MTransE	17.2	48.4	0.274	19.4	52.8	0.304	18.5	52.7	0.297
RSNs	51.9	76.4	0.606	52.0	76.1	0.605	56.5	81.8	0.653
MuGNN	36.1	72.6	0.478	30.6	69.4	0.431	35.5	74.8	0.483
AliNet	41.1	67.6	0.507	44.0	69.6	0.532	43.3	72.3	0.536
KECG	43.9	80.4	0.560	45.5	82.1	0.577	45.3	82.3	0.577
ITransE	14.9	41.2	0.236	19.7	47.7	0.290	16.9	45.5	0.264
BootEA	57.4	81.7	0.657	55.5	81.4	0.641	59.2	84.6	0.678
NAEA	30.3	55.7	0.390	28.2	54.8	0.370	29.7	56.7	0.388
TransEdge	58.0	84.9	0.676	54.2	82.3	0.642	57.3	87.5	0.682
MRAEA	61.7	**88.1**	0.711	62.9	89.1	0.722	63.6	90.7	0.738
GCN-Align	40.1	73.7	0.516	39.0	73.3	0.507	37.5	74.7	0.500
JAPE	36.8	70.1	0.481	32.7	65.0	0.435	27.3	62.1	0.390
HMAN	55.6	85.0	0.659	55.5	86.0	0.662	54.2	87.2	0.658
RDGCN	69.2	83.5	0.743	75.1	87.5	0.795	87.6	95.0	0.903
HGCN	70.7	85.3	0.759	74.6	88.0	0.794	87.1	95.3	0.901
GM-Align	55.2	76.2	0.630	61.1	81.2	0.686	76.8	92.5	0.829
CEAFF w/o C	**73.6**	86.9	**0.785**	**80.3**	**91.1**	**0.842**	**93.3**	**98.1**	**0.951**

[1]H@1 and H@10 represent Hits@1 and Hits@10, respectively.

Table 5. Results of CEAFF and PARIS on EA Datasets

	$DBP15K_{ZH-EN}$			$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$			$SRPRS_{EN-FR}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CEAFF	0.81	0.81	0.81	0.87	**0.87**	0.87	0.97	**0.97**	**0.97**	0.97	**0.97**	**0.97**
PARIS	**0.88**	**0.87**	**0.88**	**0.98**	0.84	**0.90**	**0.99**	0.93	0.96	**0.99**	0.87	0.93
	$SRPRS_{EN-DE}$			$SRPRS_{DBP-WD}$			$SRPRS_{DBP-YG}$			DBP–FB		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CEAFF	0.98	**0.98**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.96	**0.96**	**0.96**
PARIS	**0.99**	0.93	0.96	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**	0.56	0.71

P, R represent precision and recall, respectively.

representative collective ER approach, PARIS [42]. Built on the similarity comparison between literals, PARIS devises a probabilistic algorithm to jointly align entities by leveraging the KG structure and attributes.[6]

We use precision, recall and F1 score as the evaluation metrics. Different from state-of-the-art EA methods, PARIS does not necessarily generate a target entity for each source entity, and hence its precision, recall, and F1 score values could be different.

As shown in Table 5, overall speaking, PARIS achieves high precision while relatively low recall, since it merely generates matches that it believes to be highly confident. In terms of F1 score, CEAFF outperforms PARIS on most datasets. On $DBP15K_{ZH-EN}$ and $DBP15K_{JA-EN}$, PARIS attains better

[6]PARIS can also align relations and deal with unmatchable entities, while these are not the focus of the EA task.

Table 6. Time Costs of CEA and CEAFF (in Seconds)

Method	Metric	DBP15K			SRPRS				DBP-FB
		ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	DBP-WD	DBP-YG	
CEA	/	285.7	187.5	109.9	102.7	110.0	112.6	94.6	615.3
CEAFF	Mean	3,757.1	3,064.1	759.0	832.8	532.6	164.9	132.9	506.6
	Median	3,828.0	3,059.2	761.0	833.6	539.6	166.3	137.0	503.9
	Percentile 95	3,837.1	3,079.8	762.8	847.4	540.2	172.2	138.1	512.3

Table 7. The Precision Results of Ablation Study and Feature Analysis

	DBP15K			SRPRS				DBP-FB	AVG
	ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	DBP-WD	DBP-YG		
CEAFF	**0.811**	**0.868**	**0.972**	**0.968**	**0.981**	**1.000**	**1.000**	**0.963**	**0.945**
w/o C	0.736	0.803	0.933	0.931	0.946	**1.000**	**1.000**	0.814	0.895
w/o AFF	0.800	**0.868**	0.971	0.966	0.980	**1.000**	**1.000**	0.957	0.943
w/o $\boldsymbol{M^s}$	0.657	0.748	0.938	0.935	0.955	**1.000**	**1.000**	0.751	0.873
w/o $\boldsymbol{M^n}$	0.471	0.507	0.950	0.957	0.972	**1.000**	**1.000**	0.913	0.846
w/o $\boldsymbol{M^l}$	0.801	0.860	0.939	0.742	0.839	**1.000**	**1.000**	0.901	0.885
w/o θ_1, θ_2	0.799	0.863	**0.972**	0.967	**0.981**	**1.000**	**1.000**	0.958	0.943
CEAFF-cosine	0.784	0.854	0.966	0.967	0.978	**1.000**	**1.000**	0.956	0.938
CEAFF-Manh	0.733	0.794	0.953	0.941	0.962	**1.000**	**1.000**	0.919	0.907
CEAFF-Euc	0.787	0.850	0.965	0.963	0.972	**1.000**	**1.000**	0.939	0.934
AFF	0.736	0.803	0.933	0.931	0.946	**1.000**	**1.000**	0.814	0.895
LR	0.738	0.801	0.937	0.919	0.934	**1.000**	**1.000**	0.812	0.893
LM	0.665	0.727	0.915	0.909	0.932	**1.000**	**1.000**	0.798	0.868

results, as the additional attribute information it considers can provide more useful signals for alignment.

Running Time Comparison. We compared the time cost of CEAFF and CEA. Since there is randomness in each run of CEAFF, we conducted the experiments on the same dataset split for ten times and reported the averaged time cost, as well as the median and percentile 95 value of the time cost distribution. It reads from Table 6 that CEA is more efficient than our proposed RL model on most datasets. This is because: (1) Although CEA has the worst time complexity of $O(n^2)$, empirically, most source entities can find the match in a few rounds given the reasonably accurate similarity matrix; (2) The time cost of CEAFF is influenced by the number of epochs. A larger number of epochs leads to a more stable alignment result, but it also costs more time. Notably, CEA is slower than CEAFF on the lager dataset DBP-FB, which suggests that the runtime of CEA grows quickly when the scale of dataset increases. Besides, it should be noted that CEAFF is more efficient than most of the other state-of-the-art methods (whose runtime costs are shown in Table 6 in Reference [61]).

8.2 Usefulness of Proposed Features

We conduct an ablation study to gain insight into the components of CEAFF, and the results are presented in Table 7. C represents the collective alignment strategy, AFF denotes the adaptive feature fusion strategy, $\boldsymbol{M^s}, \boldsymbol{M^n}, \boldsymbol{M^l}$ represent the structural, semantic, and string-level features,

Table 8. The Precision Results of Alignment by Using Different Distance Measures

	Structure	Name	Comb.	Structure	Name	Comb.	Structure	Name	Comb.
	ZH-EN			JA-EN			FR-EN		
BC	**0.369**	**0.597**	**0.730**	**0.381**	**0.669**	**0.801**	**0.369**	**0.822**	**0.930**
Cosine	0.333	0.584	0.707	0.343	0.656	0.778	0.325	0.811	0.927
Manhattan	0.355	0.596	0.680	0.356	0.664	0.740	0.355	0.821	0.911
Euclidean	0.354	0.584	0.719	0.353	0.656	0.781	0.346	0.811	0.928
	EN-FR			DBP-WD			DBP-FB		
BC	**0.242**	0.524	**0.930**	**0.270**	**1.000**	**1.000**	**0.171**	0.568	**0.810**
Cosine	0.196	**0.530**	**0.930**	0.264	**1.000**	**1.000**	0.149	**0.583**	0.775
Manhattan	0.224	0.515	0.904	0.253	**1.000**	**1.000**	0.158	**0.583**	0.752
Euclidean	0.222	0.426	0.920	0.252	**1.000**	**1.000**	0.156	**0.583**	0.807

The results on $\text{SRPRS}_{\text{DBP-YG}}$ and $\text{SRPRS}_{\text{EN-DE}}$ are omitted in the interest of space, which are similar to $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{EN-FR}}$, respectively.

respectively. CEAFF-cosine, CEAFF-Manh, CEAFF-Euc denote replacing the Bray-Curtis dissimilarity with the cosine similarity, Manhattan distance and Euclidean distance, respectively. Notably, on $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{DBP-YG}}$, merely using semantic or string-level information can already achieve the precision of 1. Therefore, removing most components does not hurt the performance on these two datasets. In the following analysis, unless otherwise specified, we do not discuss the ablation results on $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{DBP-YG}}$.

To address RQ2, in this subsection, we examine the usefulness of our proposed features (CEAFF vs. CEAFF -$\boldsymbol{M}^s$, $\boldsymbol{M}^n$, $\boldsymbol{M}^l$). As shown in Table 7, removing the structural information leads to lower alignment results. Besides, the semantic information plays a more important role on KGs with distantly-related language pairs, e.g., $\text{DBP15K}_{\text{ZH-EN}}$, while the string-level feature is significant for aligning KGs with closely related language pairs, e.g., $\text{SRPRS}_{\text{EN-FR}}$.

8.3 Influence of Distance Measures

In this subsection, we address RQ3. As can be observed from Table 7, replacing the Bray-Curtis dissimilarity with other distance measures brings down the performance (CEAFF vs. CEAFF-cosine, CEAFF-Manh, CEAFF-Euc), which validates that an appropriate distance measure can better capture the similarity between entities.

Then, we remove the influence of the collective alignment and adaptive feature fusion strategies, and directly compare the performance of these four distance measures. As depicted in Table 8, **Structure** denotes merely using structural embeddings for alignment, **Name** denotes merely using entity name embeddings for alignment, while **Comb.** refers to fusing structural, semantic and string information with equal weights. The performance of solely using string similarity is omitted, since it is not influenced by distance measures.

It shows in Table 8 that, using the Bray-Curtis dissimilarity (BC) brings the best performance on all datasets in terms of **Structure** and **Comb**. Regarding the **Name** category, the Bray-Curtis dissimilarity also attains the best results on most datasets. This further verifies the effectiveness of this distance measure.

8.4 Effectiveness of the Adaptive Feature Fusion Strategy

In this subsection, we proceed to answer RQ4 and examine the contribution of the adaptive feature fusion strategy (CEAFF vs. CEAFF w/o AFF in Table 7). Specifically, we replace the dynamic weight assignment with fixed weights, i.e., the same weight for each feature. As can be observed

from Table 7, overall speaking, using the adaptive feature fusion strategy improves the results of using the fixed weighting, validating its usefulness. It is also noted that the increment is not very significant. This is because, by treating different features equally, the averaged weighting is the safest and the most common approach for feature fusion. Although our proposed adaptive feature fusion strategy can optimize the assignment of weights adaptively, it cannot change the input features, and hence cannot bring substantial improvement.

Thresholds θ_1, θ_2 in Adaptive Feature Fusion. As mentioned in Section 5, for a correspondence with very large similarity score, i.e., exceeding θ_1, we set its weight to a small value θ_2. To examine the usefulness of this strategy, we report the results after removing this setting in Table 7. Without this setting, the performance drops on most datasets, verifying its effectiveness.

The Learning-based Fusion Strategy. Our adaptive feature fusion strategy (**AFF)** can dynamically determine the weights of features without training data. For the comprehensiveness of the experiment, we compare with stronger baselines with learnable parameters in terms of aggregating features for alignment. We first adopt the Logistic Regression algorithm (LR) to determine the weights of features by casting the alignment to the classification problem, i.e., labeling correct EA pairs with "1"s and false pairs with "0"s. We use the learned weights to combine features and rank the target entities according to the fused similarity matrix. Note that we do not employ the collective alignment strategy to exclude its influence on the results. Besides, we also adopt a learning-to-rank algorithm, LambdaMART (LM) [9], to learn how to directly aggregate features and rank the target entities without generating the weights of features.

To construct a more useful training set, for each positive pair, we generate the negative samples by replacing the gold target entity with an incorrect target entity from the set of top-10 ranked target entities produced by solely using the structural, semantic or string-level feature, respectively. In this way, the learned models can be more discriminative compared with using randomly selected negative samples. Notably, since the original training set is used as the seed to unify the individual structural embedding spaces, the entities in the entity pairs of the training set have very high structural similarity. Therefore, it is inappropriate to still use the original training set to learn how to combine the features and rank the target entities (in which case the structural feature would be "favored" and considered as extremely useful). Instead, we use the validation set to train the models. We use the default hyper-parameters of these two models, since there is no extra data for parameter optimization. For LM, we adopt Precision@1 as the metric to optimize on the training data.

The results are reported in Table 7. It reads that, the performance of LM is not promising, which, to a large extent, can be attributed to the lack of training data (i.e., only 900 pairs are used for training, while 10,500 for testing). The effectiveness of LR is also constrained by the lack of training data. Nevertheless, it achieves better results than LM and even outperforms AFF on $\text{DBP15K}_{\text{ZH-EN}}$ and $\text{DBP15K}_{\text{FR-EN}}$. Overall speaking, the performance of AFF is better than LR and LM, and it does not need training data.

8.5 Effectiveness of the RL-based Collective Strategy

In this subsection, we seek to answer RQ5. After removing the collective strategy (CEAFF vs. CEAFF w/o C), the performance drops on all cross-lingual datasets and DBP−FB, revealing the significance of considering the interdependence between EA decisions. Moreover, it observes from Tables 2 and 3 that, simultaneously modeling the *exclusiveness* and *coherence* under the RL framework consistently outperforms the alternative strategies (CEAFF vs. CEAFF-SM, CEAFF-Excl, CEAFF-Coh).

Table 9. Analysis of the Collective Alignment Constraints

Metric	Method	DBP15K		SRPRS			DBP-FB
		ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	
Precision	CEAFF	0.811	0.868	0.972	0.968	0.981	0.963
	CEAFF-SM	0.806	0.866	0.971	0.966	0.979	0.962
	CEAFF w/o C	0.736	0.803	0.933	0.931	0.946	0.814
# *MulSE*	CEAFF	1116	768	176	173	97	307
	CEAFF-SM	0	0	0	0	0	0
	CEAFF w/o C	3778	3061	1142	1157	983	4308
# *MulTE*	CEAFF	423	307	87	77	43	134
	CEAFF-SM	0	0	0	0	0	0
	CEAFF w/o C	1441	1272	483	490	437	1130

MulSE denotes the number of source entities that are matched to the same target entities. # *MulTE* denotes the number of target entities that are assigned multiple times. The analysis on $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{DBP-YG}}$ are omitted as the precision is 1 for all methods.

Table 10. The Number of Source Entities That are Matched By the Preliminary Treatment (Two Rounds) and the Percentage of Correctly Aligned Entities (PoC)

	DBP15K			SRPRS				DBP-FB
	ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	DBP-WD	DBP-YG	
Number	8,013	8,783	10,076	10,017	10,198	10,500	10,500	16,781
PoC	94.4%	96.0%	99.1%	99.2%	99.4%	100.0%	100.0%	98.3%

Analysis of Collective Alignment Constraints. We make a detailed analysis of the collective alignment constraints. It can be seen from Table 9 that, without exerting any constraint on the alignment results, the performance of CEAFF w/o C is much worse than that of the collective alignment methods CEAFF and CEAFF-SM. Besides, in its matching results, there are many source entities that are aligned to the same target entities, e.g., around 30% of the source entities on $\text{DBP15K}_{\text{ZH-EN}}$ and $\text{DBP15K}_{\text{JA-EN}}$. This could restrain its performance, since the ground-truth results in current EA benchmarks are 1-to-1 mappings.

By exerting the 1-to-1 constraint, CEAFF-SM improves the performance, and produces a 1-to-1 mapping result (there are no *MulSE* and *MulTE*). Nevertheless, as illustrated in Example 2, the 1-to-1 constraint could cause the error propagation issue. In this work, we use the exclusiveness constraint to relax the 1-to-1 constraint and use the coherence constraint to keep the alignment decisions coherent. The resulting model CEAFF attains better performance than CEAFF-SM and CEAFF w/o C. Additionally, as shown in Table 9, employing the exclusiveness constraint allows a small number of source entities to match with the same target entities, which can partially mitigate the error propagation issue.

Influence of Preliminary Treatment. We further analyze the effectiveness of the preliminary treatment of the RL model, i.e., filtering out the source entities that can be correctly aligned by merely using the fused similarity scores. We first report the number of source entities that are matched by the preliminary treatment, as well as the percentage of the correctly aligned ones, in Table 10. The results prove that our preliminary treatment strategy can filter out many source entities, among which most are correctly aligned.

Then, we examine: (1) whether the preliminary treatment contributes to the performance of the RL-based alignment model; and (2) the appropriate rounds for conducting the filtering process.

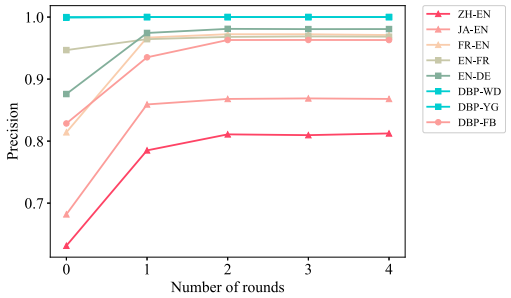


Fig. 5. Precision of CEAFF with different rounds of the preliminary treatment.

Figure 5 indicates that, this filtering process does improve the alignment performance, because: (1) it prevents the source entities that can already be confidently aligned using similarity scores from being misled by the inaccurate signals during the RL process; (2) it reduces the learning steps in each episode and leads to faster convergence; and (3) the confident matches detected by the preliminary treatment can provide more accurate coherence information for aligning the rest of the source entities. Additionally, applying this treatment for two rounds is enough, since the filtering process cannot generate 100% correct confident matches on most datasets. More rounds of the preliminary treatment increase the errors, which in turn could hurt the overall performance.

8.6 Error Analysis and Case Study

Finally, we turn to RQ6. We perform error analysis and case study to examine the cases where CEAFF falls short.

First, we analyze the change of the error rate after adding our proposed components. Take $\text{DBP15K}_{\text{ZH-EN}}$ for example. Solely using structural information leads to 63.1% error rate of the precision. Incorporating the entity name to complement the structural information reduces the overall error rate to 27%. After applying the adaptive feature fusion strategy, the error rate drops to 26.4%. On top of it, using the RL framework to collectively align entities leads to an error rate of 18.9%. This implies that, all the components in our model contribute to the alignment performance.

Nevertheless, we note that CEAFF still fails to generate correct target entities for some source entities, especially on $\text{DBP15K}_{\text{ZH-EN}}$ and $\text{DBP15K}_{\text{JA-EN}}$. Therefore, we carefully analyze the erroneous matches and broadly divide them into five categories:

Category 1. This group of cases are misled by the structural information, e.g., the match (*Saison 26 des Simpson*, *The Simpsons (season 23)*). Using the entity name information, it is easy to align the correct target entity *The Simpsons (season 26)* to the source entity. Nevertheless, the target entity *The Simpsons (season 23)* shares more common neighboring entities with the source entity *Saison 26 des Simpson*, and hence is falsely considered as the corresponding target entity.

Category 2. This group of cases are misled by the textual information, e.g., the match (*Sion (Valais)*, *Valais*). By merely using the structural information, the correct target entity *Sion,_Switzerland* can be aligned to *Sion (Valais)*. However, the high textual similarity between (*Sion (Valais)*, *Valais*) leads to the wrong match, despite of their structural disparity.

Category 3. Cases in this group are misled by the feature fusion strategy, e.g., the match (*Hérouxville*, *Hérault*). It is noted that using structural information or entity name information can find the correct target entity *Hérouxville,_Quebec*, while combining these features leads to the wrong result.

Table 11. The Percentages of Different Types of Errors Made By CEAFF

	DBP15K			SRPRS		DBP-FB	AVG
	ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE		
Error Rate	18.9%	13.2%	2.8%	3.2%	1.9%	3.7%	7.3%
Category 1	2.7%	4.7%	15.6%	14.8%	12.8%	11.4%	10.3%
Category 2	17.8%	16.4%	15.6%	8.3%	10.7%	3.2%	12.0%
Category 3	0.1%	0.1%	6.3%	0.6%	0.0%	0.0%	1.2%
Category 4	11.1%	14.1%	18.9%	15.1%	17.3%	15.6%	15.4%
Category 5	68.3%	64.7%	43.6%	61.2%	59.2%	69.8%	61.1%

Category 4. The collective alignment strategy is responsible for this category of errors, e.g., the match (*Grande Ourse*, *Landrienne,_Quebec*). Based on the fused similarity matrix, generating the results independently can find the correct target entity *Ursa Major* for the source entity. Nevertheless, the target entity *Ursa Major* has a higher similarity score with another source entity *Bouvier (constellation)*. As thus, when aligning entities collectively, the *exclusiveness* constraint prevents *Grande Ourse* from aligning *Ursa Major*.

Category 5. Cases in this group cannot be tackled by any module in our model, e.g., the match (*Yolande de Hongrie (reine d'Aragon)*, *Constance of Sicily,_Queen of Aragon*). This might be ascribed to the fact that the correct target entity *Violant of Hungary* neither has a similar name, nor shares similar structural information, with the source entity.

We report the percentages of these categories of errors in Table 11.[7] Next, we analyze the errors and suggest some possible research directions in the future.

It can be observed from Table 11 that, Categories 1 and 2 account for 10.3% and 12% on average, respectively. This shows that there is still room for improving the feature encoders to learn better representations. The feature fusion strategy (Category 3), however, brings few errors. 15.4% of the incorrect matches are caused by the collective alignment strategy (Category 4). Admittedly, aligning entities jointly can better model the correlations between EA decisions, and hence significantly improves the alignment results. Nevertheless, an erroneous match can trigger error propagation and leads to more erroneous matches, as shown in the example of Category 4. Therefore, more advanced collective alignment strategies could be devised to further mitigate this issue. Notably, Category 5 takes the largest share, revealing that the majority of errors are not generated by the components of our model. However, these errors might be avoided by mining more useful features or designing more advanced approaches to exploit available features.

9 CONCLUSION

When making EA decisions, current EA solutions treat entities separately, or fail to adequately model the interdependence among entities. To fill in this gap, we frame EA as the sequence decision process and devise a deep RL model to capture both the exclusiveness and coherence of EA decisions. Besides, we put forward an adaptive feature fusion strategy to aggregate multiple features and provide more accurate inputs to the RL framework. Compared with state-of-the art approaches, our proposal achieves consistently better results, and the ablation study also verifies the usefulness of each component. More concretely, the results show that: (1) aligning entities collectively using reinforcement learning can sufficiently capture the interdependence between EA decisions; (2) dynamically fusing the features with adaptively assigned weights can better take

[7]The results on $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{DBP-YG}}$ are omitted, since they do not generate erroneous matches.

into consideration the strength of each feature compared with equal weights; and (3) compared with existing distance measures, the Bray-Curtis dissimilarity is a better measure to characterize the distance between entity embeddings.

We will explore the following directions in the future: (1) devising more advanced feature encoders that can better exploit available features for alignment; (2) designing collective alignment algorithms that can further mitigate the error propagation caused by the erroneous matches; and (3) establishing a more challenging (mono-lingual) EA benchmark.

APPENDIX

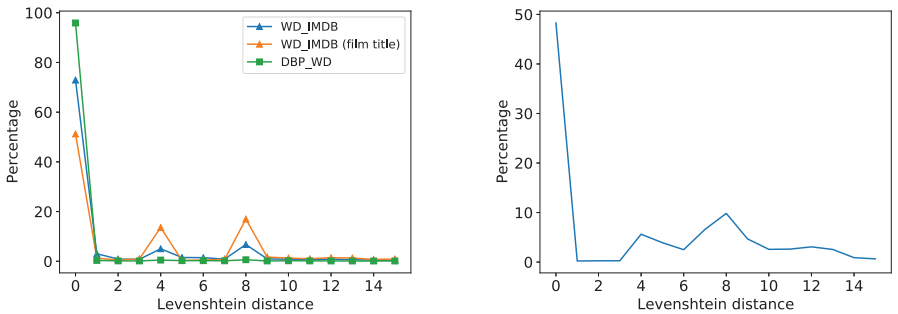
A FURTHER ANALYSIS OF THE STRING FEATURE

This Appendix provides more detailed analysis of the string feature. First, we report the average, median, percentile-10, and percentile-90 Levenshtein distance between the names of the gold entity pairs in each dataset in Table 12. It can be observed that the string feature is extremely useful on $\text{SRPRS}_{\text{DBP-WD}}$ and $\text{SRPRS}_{\text{DBP-YG}}$. This is because these two datasets are extracted from DBpedia, Wikidata and YAGO, where equivalent entities in different KGs possess identical labels, and a simple comparison of these labels can achieve ground-truth results [61].

Table 12. Statistics of the Levenshtein Distance Between the Names of the Gold Entity Pairs in Each Dataset

	DBP15K			SRPRS				DBP-FB
	ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	DBP-WD	DBP-YG	
Average	13.8	14.5	5.2	3.3	3.1	0	0	4.8
Median	13	14	0	0	0	0	0	4
Percentile-90	23	24	16	13	12	0	0	12
Percentile-10	6	7	0	0	0	0	0	0

We further investigate into the string feature by creating a new dataset (WD_IMDB) with KGs from independent sources (Wikidata and IMDB) and comparing the distribution of the Levenshtein distance between the entity names of gold entity pairs on DBP_WD and WD_IMDB.



(a). On DBP_WD and WD_IMDB (b). On DBP_FB

Fig. 6. Distribution of the Levenshtein distance between the names of gold entity pairs.

More specifically, we created the Wikidata-IMDB KG pair, WD_IMDB, as well as the WD_IMDB (film title) dataset that only comprises film entities. Figure 6(a) shows the the distribution of the Levenshtein distance between the names of the gold entity pairs in WD_IMDB, WD_IMDB (film title) and DBP_WD. It reads that on WD_IMDB, the distribution of Levenshtein distance is relatively

more "normal" than that on DBP_WD. Besides, there is a less portion of entities that have exactly the same labels on WD_IMDB (film title) compared with WD_IMDB. This is because many entities in WD_IMDB are film-related persons, and the names of persons tend to be the same in different KGs, while the titles of film entities are more likely to be described differently.

In summary, it can be concluded that the name information is indeed causing overfitting issue on DBpedia, Wikidata and YAGO. As a consequence, we adopt a recently constructed dataset DBP-FB, which also mitigates the overfitting issue of the string feature, to evaluate state-of-the-art methods. The distribution of the Levenshtein distance over DBP-FB is shown in Figure 6(b), which is similar to WD_IMDB (film title). We reckon DBP-FB is a more appropriate dataset for EA compared with WD_IMDB, since IMDB merely contains entities in a few types (mostly films and persons) and is domain-specific, while Freebase is a more general KG. In consequence, we include the DBP-FB dataset in the experiment and evaluate the methods mentioned in this article on it.

B FURTHER EXPERIMENT ON CONFIDENT CORRESPONDENCE GENERATION

As introduced in Section 5.2, an important step in our adaptive feature fusion strategy is to detect confident correspondence. We report the percentage of correctly generated confident correspondences in Table 13. It shows that our proposed confident correspondence generation strategy indeed leads to a high percentage of correct correspondences.

Table 13. The Percentage of Correctly Generated Confident Correspondences (PoC)

	DBP15K			SRPRS				DBP-FB
	ZH-EN	JA-EN	FR-EN	EN-FR	EN-DE	DBP-WD	DBP-YG	
PoC	85.6%	87.1%	90.1%	89.0%	93.5%	93.3%	93.3%	82.6%

REFERENCES

[1] Yasser Altowim, Dmitri V. Kalashnikov, and Sharad Mehrotra. 2014. Progressive approach to relational entity resolution. *Proc. Endow. Very Large Data Base* 7, 11 (2014), 999–1010.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of ISWC*. 722–735.

[3] Indrajit Bhattacharya and Lise Getoor. 2006. A latent dirichlet model for unsupervised entity resolution. In *Proceedings of ICDM*. 47–58.

[4] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *Trans. Knowl. Discov. Data* 1, 1 (2007), 5. DOI: https://doi.org/10.1145/1217299.1217304

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* 5 (2017), 135–146.

[6] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*, Jason Tsong-Li Wang (Ed.). ACM, 1247–1250.

[7] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*. 2787–2795.

[8] J. Roger Bray and John T. Curtis. 1957. An ordination of the upland forest communities of southern Wisconsin. *Ecol. Monogr.* 27, 4 (1957), 325–349.

[9] Christopher J. C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report. Microsoft Research. Retrieved from http://research.microsoft.com/en-us/um/people/cburges/tech_reports/MSR-TR-2010-82.pdf.

[10] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of ACL*. 1452–1461.

[11] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of WWW*. 151–161.

[12] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training Embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of IJCAI*. 3998–4004.

[13] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*. 1511–1517.
[14] Peter Christen. 2012. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Trans. Knowl. Data Eng.* 24, 9 (2012), 1537–1555. DOI: https://doi.org/10.1109/TKDE.2011.127
[15] Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of EMNLP*. 2256–2262.
[16] Sanjib Das, Paul Suganthan G. C., AnHai Doan, Jeffrey F. Naughton, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, Vijay Raghavendra, and Youngchoon Park. 2017. Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In *Proceedings of SIGMOD*. 1431–1446.
[17] Jack Doerner, David Evans, and Abhi Shelat. 2016. Secure stable matching at scale. In *Proceedings of SIGSAC*. 1602–1613.
[18] Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning. In *Proceedings of WWW*. 438–447.
[19] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI, IAAI, and EAAI*. 5779–5786.
[20] Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. End-to-end multi-perspective matching for entity resolution. In *Proceedings of IJCAI*. 4961–4967.
[21] David Gale and Lloyd S. Shapley. 1962. College admissions and the stability of marriage. *Amer. Math. Month.* 69, 1 (1962), 9–15.
[22] Marko Gulic, Boris Vrdoljak, and Marko Banek. 2016. CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. *J. Web Semant.* 41 (2016), 50–71.
[23] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of ICML*. 2505–2514.
[24] Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of NAACL-HLT*. 851–861.
[25] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.
[26] Pigi Kouki, Jay Pujara, Christopher Marcum, Laura M. Koehly, and Lise Getoor. 2019. Collective entity resolution in multi-relational familial networks. *Knowl. Info. Syst.* 61, 3 (2019), 1547–1581.
[27] Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Res. Logist. Quart.* 2, 1–2 (1955), 83–97.
[28] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. SIGMa: Simple greedy matching for aligning large knowledge bases. In *Proceedings of KDD*. 572–580.
[29] Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Phys. Doklady*, Vol. 10. 707–710.
[30] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *Proceedings of EMNLP-IJCNLP*. 2723–2732.
[31] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: An efficient and robust entity alignment approach for cross-lingual knowledge graph. In *Proceedings of WSDM*. 420–428.
[32] Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of NIPS*. 905–912.
[33] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of ICML*. 1928–1937.
[34] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of SIGMOD*. 19–34.
[35] Hao Nie, Xianpei Han, Ben He, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. Deep sequence-to-sequence entity matching for heterogeneous entity resolution. In *Proceedings of CIKM*. 629–638.
[36] Ning Pang, Weixin Zeng, Jiuyang Tang, Zhen Tan, and Xiang Zhao. 2019. Iterative entity alignment with improved neural attribute embedding. In *Proceedings of DL4KG@ESWC*. 41–46.
[37] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* 8, 3 (2017), 489–508.
[38] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proceedings of WWW*. 3130–3136.
[39] Alvin E. Roth. 2008. Deferred acceptance algorithms: History, theory, practice, and open questions. *Int. J. Game Theory* 36, 3–4 (2008), 537–569.

[40] Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* 27, 2 (2015), 443–460.
[41] Parag Singla and Pedro M. Domingos. 2006. Entity resolution with markov logic. In *Proceedings of the ICDM*. 572–582.
[42] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. PARIS: Probabilistic alignment of relations, instances, and schema. *Proc. Endow. Very Large Data Base* 5, 3 (2011), 157–168.
[43] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of WWW*. 697–706.
[44] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of ISWC*. 628–644.
[45] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of IJCAI*. 4396–4402.
[46] Zequn Sun, JiaCheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. TransEdge: Translating relation-contextualized embeddings for knowledge graphs. In *Proceedings of ISWC*. 612–629.
[47] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of EAAI*. 222–229.
[48] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of AAAI*. 297–304.
[49] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
[50] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*. 349–357.
[51] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8 (1992), 229–256.
[52] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of IJCAI*. 5278–5284.
[53] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly learning entity and relation representations for entity alignment. In *Proceedings of EMNLP-IJCNLP*. 240–249.
[54] Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of WWW*. 1271–1279.
[55] Kun Xu, Linfeng Song, Yansong Feng, Yan Song, and Dong Yu. 2020. Coordinated reasoning for cross-lingual knowledge graph alignment. In *Proceedings of AAAI*. 9354–9361.
[56] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual knowledge graph alignment via graph matching neural network. In *Proceedings of ACL*. 3156–3161.
[57] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *Proceedings of EMNLP-IJCNLP*. 4430–4440.
[58] Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2020. Collective entity alignment via adaptive features. In *Proceedings of ICDE*. IEEE, 1870–1873.
[59] Weixin Zeng, Xiang Zhao, Wei Wang, Jiuyang Tang, and Zhen Tan. 2020. Degree-aware alignment for entities in tail. In *Proceedings of SIGIR*. ACM, 811–820.
[60] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of IJCAI*. 5429–5435.
[61] Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian Suchanek. 2020. An experimental study of state-of-the-art entity alignment approaches. *IEEE Trans. Knowl. Data Eng.* (2020), 1–1. https://ieeexplore.ieee.org/document/9174835.
[62] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of IJCAI*. 4258–4264.
[63] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of IJCAI*. 1943–1949.

Received June 2020; revised November 2020; accepted December 2020