Electronic
Research Archive

https://www.aimspress.com/journal/era

ERA, 33(8): 4559–4602.
DOI: 10.3934/era.2025206
Received: 22 May 2025
Revised: 07 July 2025
Accepted: 16 July 2025
Published: 07 August 2025

Research article

An multi-embedding entity alignment approach based on matching information interaction over atomically divisible entities

Yaoli Xu, Tong Han, Pu Li*, Changhai Wang*, Xiaoyu Duan* and Haojie Zhai

College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

* **Correspondence:** Email: lipu@zzuli.edu.cn, chw@zzuli.edu.cn, dxyuniv@zzuli.edu.cn.

Abstract: Entity alignment, as a fundamental technique in knowledge graph integration, aims to identify the equivalent entity pairs that refer to the same real-world entity across different knowledge graphs. In domain-specific knowledge graphs, based on the atomic divisibility characteristics of entity categories, these categories can be classified into atomically divisible and atomically indivisible types. However, traditional methods for entity alignment in atomically indivisible knowledge graphs often fall short when dealing with tasks that involve more than one atomically divisible entity category. To address such a challenge, we propose a novel multi-embedding entity alignment approach based on matching information interaction over atomically divisible entities (MiAD). MiAD effectively exploits the matching information interaction between low-level entities and high-level entities to enrich the semantic representation of high-level entities and to enhance the reliability of the alignment results. The high-level entities are atomically divisible, and the matching result among the low-level entities can provide a significant hint for the entity alignment of high-level entities by analyzing their inherent interrelationships. To tackle the insufficient utilization of these matching information interactions, we propose a multiple hierarchical embedding strategy, which consists of a structural semantic aggregator, an attribute semantic aggregator, a corroborative aggregator, and a joint aggregator. This method leverages both the intra-level interactions and cross-level interactions to generate the hierarchical embeddings of high-level and low-level entities, respectively, from multiple perspectives (i.e., relations, attributes, and corroborative textural descriptions). As a result, MiAD significantly improves the accuracy of alignment outcomes and strengthens the overall structural and semantic understanding of higher-level entities. Comprehensive experimental results demonstrate that MiAD achieves substantial performance improvements compared with seven baseline methods on the Chinese recipe dataset Tada.

Keywords: matching information interaction; atomic entity; composite entity; intra-level interaction; cross-level interaction; entity alignment

1. Introduction

A knowledge graph (KG) uses graph models to describe knowledge and model relationships between entities [1]. As a crucial cornerstone in the transition from perceptual intelligence to cognitive intelligence in the big data era, knowledge graphs excel in semantic expression, knowledge storage, and reasoning capabilities. They provide efficient solutions for applications such as knowledge reasoning [2], intelligent recommendation [3], and intelligent question answering [4, 5]. Currently, numerous general-purpose knowledge graphs have emerged both domestically and internationally, such as DBpedia [6], YAGO [7], Freebase [8], Wikidata [9], and BabelNet [10].

However, these general-purpose knowledge graphs cannot effectively model domain knowledge, and result in suboptimal performance regarding industry-specific recommendations and question-answering systems in fields such as food, healthcare, movies, and finance. Consequently, there is an increasing demand for knowledge graph construction across various industries, leading to the gradual establishment of domain-specific knowledge graphs. Due to inherent difficulties in data circulation, customer privacy, trade secrets, and regulatory requirements, data silos have emerged. Additionally, the overall poor low quality of the data hinders the full potential of what the data can offer [11]. To address these issues, domain knowledge graph researchers urgently need to explore data fusion technologies, aimed at breaking down data barriers, promoting information sharing, and improving data quality to unlock the potential of data elements.

Due to the heterogeneity and incompleteness commonly present in knowledge graphs derived from different sources, there is an urgent need to integrate multiple knowledge graphs to expand the overall knowledge base and provide richer details about entities [12, 13]. Entity alignment (EA), as a crucial technology in the construction and fusion of knowledge graphs, aims to identify all entity descriptions that refer to the same real-world entity across different open-source knowledge graphs and establish equivalent link relationships in order to construct a highly unified, large-scale, high-quality knowledge graph [12, 14]. Owing to the diverse data sources of knowledge graphs, inconsistencies in the description of an entity may arise due to different modeling standards and data processing methods being employed during their construction. Therefore, entity alignment is a key technology for addressing these issues and achieving knowledge graph fusion.

With the continuous advancement of intelligent technologies such as deep learning and natural language processing, numerous entity alignment methods have been proposed in both academia and industry. However, these traditional methods still face several challenges in domain-specific knowledge graphs.

1) Most existing methods rely on general alignment models to handle various types of entities, even when these entities belong to different categories. These methods often overlook the distinctions among entity types and fall short in effectively utilizing the abundant matching information interactions that exist between high-level entities and low-level entities.

2) Existing studies primarily focus on constructing entity embedding representations from the structural semantic perspective of knowledge graphs, but not all domain knowledge graphs contain rich structural information. Additionally, the entity names, attribute values, and descriptive information in knowledge graphs contain abundant entity-related details. Failing to fully exploit this information may result in suboptimal entity alignment performance, thereby falling short of the data quality requirements for downstream tasks such as intelligent recommendation and

question-answering.

3) Traditional methods typically require various embeddings to be built for all entities. However, noisy entities or obviously mismatched entities consume substantial computational resources, significantly reducing the efficiency of model execution.

In response to these challenges, we propose a novel multi-embedding entity alignment approach based on matching information interaction over atomically divisible entities (MiAD). According to the concept of atomic divisibility, there is a hierarchical structure among different categories of entities, where high-level entities can be decomposed into multiple low-level entities on the basis of their inherent relationships. MiAD first leverages the hierarchical structure analyzer to identify hierarchical relationships among the entities. Then MiAD effectively exploits the iterative optimization strategy to optimize the heterogeneous entity relationship inference according to the matching information interactions (i.e., intra-level interactions and cross-level interactions). To capture comprehensive entity representations, the cross-level interaction mechanism utilizes a hierarchical embedding strategy consisting of structural aggregators, attribute aggregators, corroborative aggregators, and joint aggregators, which enrich the unified comprehensive embeddings of high-level entities through analyzing the high-confidence matched low-level entity pairs. The multi-perspective semantic embedding strategy qualifies multi-perspective information including structural triples, attribute triples, entity descriptions, and corroborative textual semantics through advanced techniques (such as graph convolutional networks, self-attention mechanisms, and pretrained language models). By integrating the matching information interactions over hierarchical entities, MiAD significantly enhances both the reliability of the alignment results and the quality of entity representations, enabling effective integration of heterogeneous domain knowledge graphs. The main contributions of our paper are as follows.

1) We propose a novel entity alignment approach based on matching information interaction over atomically divisible entities, which is suitable for addressing complex entity alignment tasks. Since entities of different atomic sizes contain mutually corroborating information, effective extraction and utilization can enhance the reliability of the entity alignment results and enrich the high-level entity structural information.

2) We propose an iterative optimization strategy that identifies high-confidence aligned entity pairs and infers new relationships through intra-layer interactions, while enhancing high-level entity representations by leveraging equivalence relationships between low-level entities through inter-layer interactions. This hierarchical information interaction mechanism achieves iterative optimization of heterogeneous entity relationship reasoning, significantly improving entity alignment performance.

3) We propose a multi-perspective semantic embedding strategy that constructs four aggregators, namely: the structural aggregator, the attribute aggregator, the corroborative aggregator, and the joint aggregator, comprehensively capturing entity semantics from the structure, attributes, corroborativer information, and intrinsic correlations among these three aspects. The structural aggregator employs multi-layer graph convolutional networks (GCNs) [15] to facilitate higher-order feature learning and propagate knowledge graphs' structural information, effectively enriching missing entity connections through indirect paths. Meanwhile, the attribute aggregator identifies matched composite entities as the shared attribute values of high-level entities, extracting comprehensive semantic features that significantly improve multi-dimensional entity representation for more accurate alignment.

4) We evaluated the effectiveness of MiAD on a domain-specific dataset (Ta-da). Our experimental

results show that MiAD consistently outperforms seven traditional methods in complex entity alignment tasks. Notably, MiAD with three hidden layers achieved the highest scores on almost all metrics, with a remarkable recall of 99.87% at a candidate set size ratio (CSSR) of 5.62%.

Organization: The rest of the paper is organized as follows. Section 2 presents the definitions of the problem, key concepts, and symbols. Section 3 presents the overall model architecture, which consists of a hierarchical structure analyzer, a blocker filter, a multiple perspective semantic encoder, spatial mapping, and a heterogeneous entity relationship and iterative optimization strategy. Section 4 conducts comprehensive experiments along with a detailed analysis of the experimental results. We discuss related work in Section 5 and conclude the paper in Section 6.

2. Preliminaries

In this section, we first introduce several key concepts in domain knowledge graphs with hierarchical structures. Then, according to whether the knowledge graphs contain composite entities, the entity alignment problems can be categorized into two types: atomic entity alignment and complex entity alignment. Finally, we summarize the related symbols.

2.1. Concepts

In practical applications, particularly with domain-specific knowledge graphs, entities often belong to multiple categories. According to the atomic divisibility characteristics of entity categories, we classify entities into atomic entities and composite entities.

Atomic entities. We refer to these atomically indivisible entities as atomic entities (e.g., e_{21} and e_{31} in Figure 1). These atomic entities are entities whose attributes contain only textual or fundamental values and do not reference another entity.

Composite entities. We refer to atomically divisible entities as composite entities (e.g., e_{11} and e_{51} in Figure 1). These composite entities have certain attribute values that serve as references to other entities. These referenced entities may be atomic entities or composite entities. In other words, composite entities may consist of atomic entities, other composite entities, or a combination of both.

If a composite entity (e.g., e_{11} in Figure 1) consists of atomic entities (e_{31} in Figure 1) or composite entities (e_{51} in Figure 1), we regard the composite entity (e_{11}) as a high-level entity, and the atomic entity (e.g., e_{31}) and the composite entity (e.g., e_{51}) are regarded as a low-level entities with respect to e_{41}. The relationship between the high-level entity and the low-level entity is regarded as a hierarchical relationship r^{HS}. For example, the hierarchical relationship between e_{11} and e_{31} is represented as a relational triple $(e_{11}, r^{\mathrm{HS}}, e_{31})$. As illustrated in Figure 1, the dashed lines indicate the hierarchical relationships between high-level and low-level entities, while solid lines indicate the relationships between entities at the same hierarchical level. Circles denote entities, while squares represent their attributes.

Hierarchical matching. Hierarchical matching, a technical strategy for aligning composite entities, aims to leverage matching information interactions to enhance entity alignment involving composite entities, in a bottom-up manner. It involves intra-level matching interactions that leverage high-confidence aligned entity pairs within the same level, and cross-level matching interactions that utilize the matching results of low-level entities associated with their corresponding high-level entities.

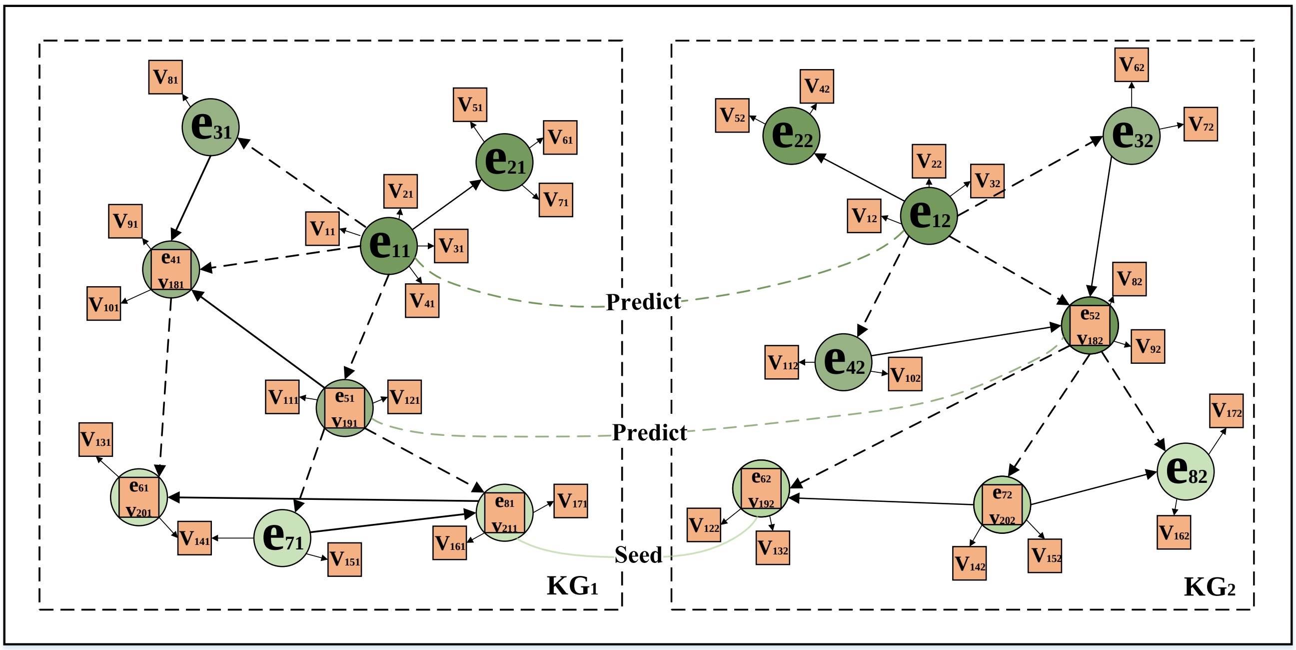

Figure 1. An illustration of a complex entity alignment problem.

2.2. *Problem definition*

In this paper, the domain knowledge graph is formulated as a quaternion, represented as $KG = (E, R, A, V)$, where E is the entity set, A is the attribute set, V is the attribute value set, and $R = R_r \cup R_a$ is the relation set, i.e., the union of the set of relational triples (R_r) and attribute triples (R_a). Additionally, $R_r = \{(h, r, t)|h, t \in E, r \in R\}$ describes relationships between entities in E, and $R_a = \{(h, a, v)|h \in E, a \in A, v \in V\}$ describes the relationships between the entities in E and their corresponding attribute values in V through the attributes in A.

Given two domain knowledge graphs $KG_1 = (E_1, R_1, A_1, V_1)$ and $KG_2 = (E_2, R_2, A_2, V_2)$, the entity alignment problem aims to determine whether an entity e_{i1} from KG_1 is equivalent to the entity e_{j2} in KG_2 or not. If e_{i1} and e_{j2} are predicted to be equivalent, the corresponding pair (e_{i1}, e_{j2}) is added to the equivalence relation set $S = \left\{(e_{i1}, e_{j2})|e_{i1} \in E_1, e_{j2} \in E_2, e_{i1} \equiv e_{j2}\right\}$, where $\equiv$ denotes the equivalence relation. Depending on whether the knowledge graphs contain composite entities, the entity alignment problem can be classified into two categories: atomic entity alignment and complex entity alignment. Importantly, when all entities are atomic, the complex entity alignment problem degenerates into the atomic entity alignment problem, which corresponds to the conventional entity alignment problem.

In practical scenarios, especially in domain knowledge graphs, both atomic and composite entities exist. The complex entity alignment problem is proposed to resolve the entity alignment task among knowledge graphs with composite entities. An illustration of the complex entity alignment problem is shown in Figure 1. The input consists of KG_1, KG_2, and S_{seed}, with each containing pairs (e.g., e_{81} and e_{62} in Figure 1) of entity descriptions that share an equivalence relationship. The output is S, the equivalence relation set that contains all pairs of entity descriptions that refer to the same real-world entity.

The atomic entity alignment problem refers to determining the equivalent relationships among the descriptions of atomic entities, as shown in Figure 2. All entities of KG_1 and KG_2 are atomic, belonging to the same classification without hierarchical relationships. Green represents entities, orange represents attributes, solid blue lines represent alignment seed pairs, and blue dashed lines represent the predicted equivalence relation.

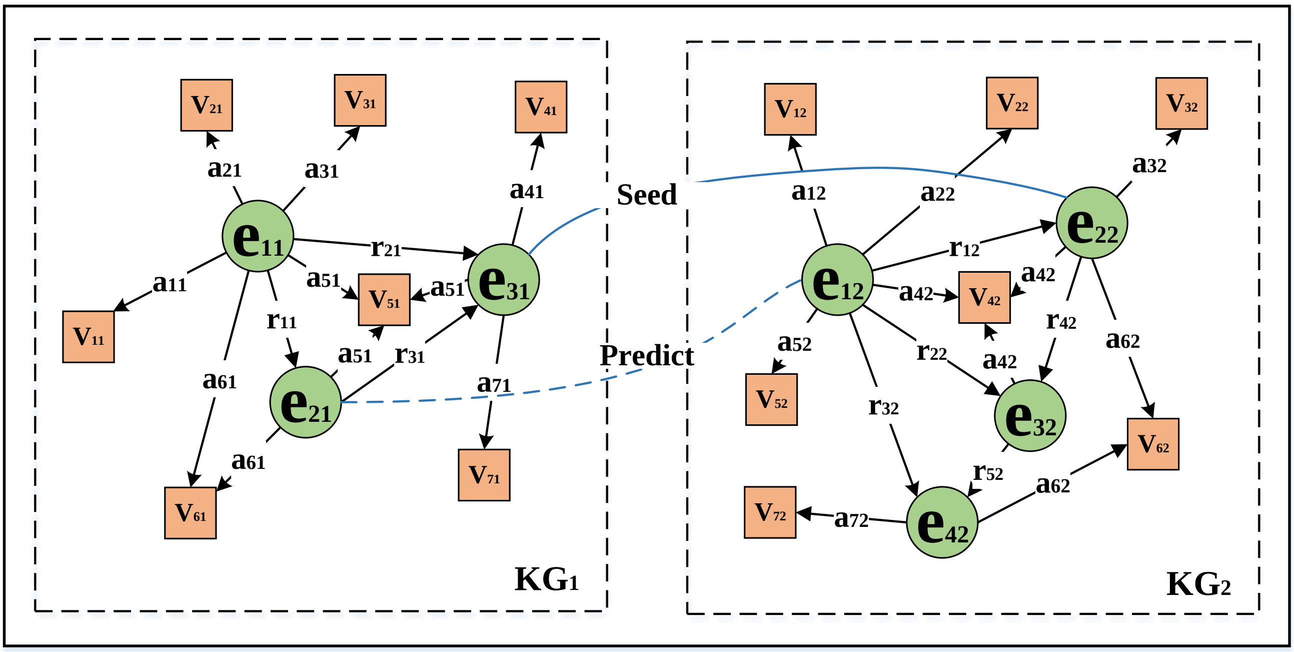

Figure 2. An illustration of an atomic entity alignment problem.

It is nontrivial to resolve the complex entity alignment problem. First, from the perspective of hierarchical relationships, entities can be classified as high-level or low-level entities. Low-level entities serve as attribute features for other high-level entities. How to effectively utilize the interaction of matching information between the entity alignment result of high-level entity descriptions and low-level entity descriptions to achieve effective matching is a nontrivial challenge. Second, unlike the atomic entity alignment problem, the complex entity alignment methods have to trade off efficiency and effectiveness when simultaneously aligning both high-level and low-level entities.

2.3. Symbols

We summarize the key symbols with their descriptions in Table 1.

Table 1. Symbols and description.

Symbol	Description
KG	The knowledge graph with complex entities $KG = (E, R, A, V)$
E, R, A, V	The set of entities, the set of relations, the set of attributes, and the set of attribute values
R_r	The set of relational triples $R_r = \{(h, r, t) \mid h, t \in E, r \in R\}$
R_a	The set of attribute triples $R_a = \{(h, a, v) \mid h \in E, a \in A, v \in V\}$
R_r^{HS}	The set of hierarchical relationships $(h_{\mathrm{i}}^{\mathrm{H}}, r^{\mathrm{HS}}, t_j^{\mathrm{L}}) \in R_r^{\mathrm{HS}}$
S	Set of equivalent entity pairs $S = \left\{\left(e_{i1}, e_{j2}\right) \mid e_{i1} \in E_1, e_{j2} \in E_2, e_{i1} \equiv e_{j2}\right\}$
S_{seed}	The seed set of aligned entity pairs
e_{i1}, e_{j2}	The entity from the knowledge graph, i.e., $e_{i1} \in KG_1$ and $e_{j2} \in KG_2$
$e_{i'1}$	The neighboring entity of entity e_{i1}
E^m	The set of entities in the m-th layer, $m \in \{0, 1, ..., M\}$
e^m	The entity existing E^m in the m-th layer
C^{K}	The top-K candidate entity set of entities
N_i	The set of neighboring entities
$Sim\left(e_{i1}, e_{j2}\right)$	The similarity between two entities $e_{i1} \in KG_1$ and $e_{j2} \in KG_2$
M	The adjacency matrix
M^d	The adjacency matrix constructed using direct structure information between entities
$M_{ii'}^d$	The direct connection between an entity e_{i1} and its neighboring entity $e_{i'1}$
M^i	The adjacency matrix constructed using the indirect structure information of entities
$M_{ii'}^i$	The indirect connection between an entity e_{i1} and its neighboring entity $e_{i'1}$
$W^{(l-1)}$	The weight matrix
$W_g^{(l-1)}$	The self-learning weight
$W_a^{(l-1)}$	The self-attention weight
$h_i^{(l)}$	The feature representation of the hidden l-th layer
$e_{ii'}$	The attention coefficient of the neighboring entity $e_{i'1}$ for entity e_{i1}
$a_{ii'}$	The attention score of the neighboring entity $e_{i'1}$ for entity e_{i1}
$\widehat{M}$	The adjacency matrix of the added self-connection relation I
$\widehat{D}$	The degree matrix
e_i^z	The entity's name
h_i^w	The character-level embedding sequence
f_i^k	The word frequency
$word_i^k$	The k-th word of the i-th entity
w_i^k	The weight of $word_i^k$
h_{G1}^n	The embedding feature vectors
$h_i^{\mathrm{s}}, h_i^{\mathrm{a}}, h_i^{\mathrm{c}}, h_i^{\mathrm{j}}$	The structural features, the attribute features, the corroborative features, and the joint features

3. Framework overview

Most of the existing methods have been proposed to deal with entities belonging to the same category. When dealing with knowledge graphs with complex entities belonging to multiple categories, the naïve method is to utilize a generic alignment model to resolve entity descriptions across all categories. There are two shortcomings in the naïve method as follows. (i) Generic alignment models often yield unsatisfactory performance, since it is hard to train a unified model for entities of each category. (ii) Multiple interactions of matching information exist among atomically divisible entities, which can offer valuable cues for enhancing the entity alignment performance.

It is observed that facilitating matching information interactions can significantly enhance the performance of entity alignment. Matching information interactions can be categorized into two types: intra-level interactions, which occur within the same type of entities, and cross-level interactions, which take place between entities of different types. The intra-level interaction contributes to improving the recall of matching results, whereas cross-level interaction plays a crucial role in enhancing the precision of matching results, since high-confidence equivalent relationships among low-level entities facilitate confident inference of matching relationships between corresponding high-level entities. Motivated by these observations, we propose MiAD—a novel multi-embedding entity alignment approach based on matching information interaction over atomically divisible entities, which fully exploits both intra-level and cross-level entity matching interactions.

As illustrated in Figure 3, MiAD consists of five key components. (i) *The hierarchy structure analyzer* is responsible for analyzing hierarchical relationships among entities by determining whether certain entities—either atomic or composite—serve as attribute values of composite entities. On the basis of this analysis, it constructs a hierarchical structure among entities of different categories, thereby establishing a foundation for matching information interactions. (ii) *The blocking filter* employs a structural feature construction strategy based on a dual-layer GCN and a similarity measure computed via the L_2-norm. By evaluating the structural similarity of entity pairs, it effectively eliminates entity pairs that are evidently nonmatching. (iii) *The multi-perspective semantic embedding strategy* is introduced to capture comprehensive representations from multiple perspectives. From the structural perspective, the structure semantic aggregator employs a dual-layer GCN enhanced by a self-attention mechanism to capture both direct and indirect structural relationships. From the attribute perspective, the attribute semantic aggregator leverages a combination of fastText and smooth inverse frequency (SIF) models to process low-level and high-level attribute information. From the corroborative information perspective, the corroborative aggregator utilizes bidirectional encoder representations from transformers (BERT) to extract the semantic features from textual descriptive information. From the perspective of inter-feature correlations, the joint aggregator integrates these high-dimensional features into a unified entity representation, aiming to capture the intrinsic correlations among features from different perspectives. (iv) *Spatial mapping and heterogeneous entity relationship inference*: Space mapping can address potential inconsistencies in the embedding spaces between knowledge graphs from different sources by applying the transformation matrix. The heterogeneous entity relationship inference model is optimized through the minimization of a modified loss function. (v) *The iterative optimization strategy* leverages a hierarchical matching information propagation mechanism, in which the optimization of the entity relationship inference module and the matching information interactions (including both intra-level and cross-level mechanisms) are performed in an iterative manner. By enabling the exchange of matching information flow across entities at different hierarchical layers, it facilitates more effective matching interactions and ultimately improves the alignment performance.

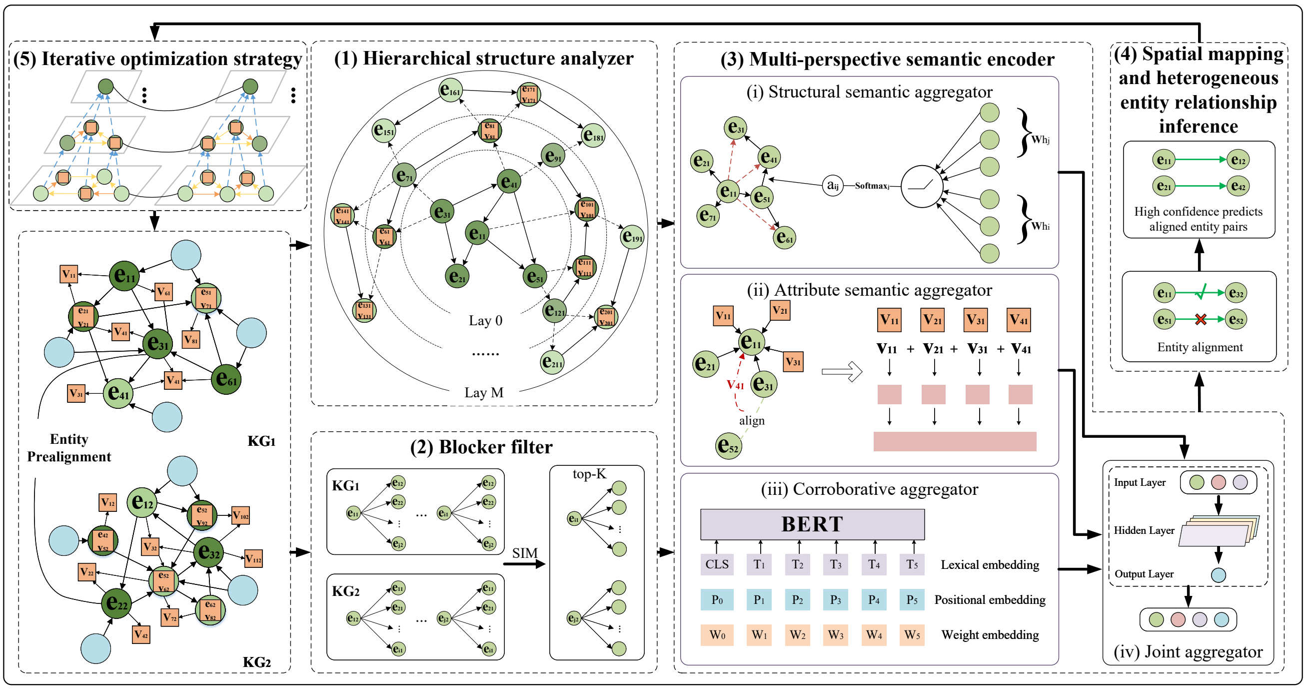

Figure 3. The architecture of MiAD.

3.1. Hierarchical structure analyzer

In many domain-specific knowledge graphs, there is a hierarchical structure among different categories of entities, where high-level entities consist of low-level entities. For instance, (i) in pharmaceutical knowledge graphs, pharmaceutical entities (e.g., Suhuang-Zhike-Jiaonang), as the high-level entities, are composed of certain low-level entities such as drug entities (e.g., ephedra and perilla leaf). As the low-level entity, ephedra consists of chemical entities (e.g., alkaloids, flavonoids, organic acids, and amino acids). (ii) In food knowledge graphs, dish entities (e.g., scrambled eggs with tomatoes) may contain subentities like the ingredient entities (e.g., tomato) and seasoning entities (e.g., salt). As the high-level entity, a tomato comprises various vitamin entities (e.g., Vitamin C). These hierarchical structures reflect the subordination and composition relationships between entities, which make it possible to interact matching information among different levels for entity alignment. Such insight is ignored by current state-of-the-art entity alignment methods. We are the first to employ the matching information of low-level entities to enhance the matching results of high-level entities.

However, given a knowledge graph with composite entities, how to identify these hierarchical relationships among different categories of entities is nontrivial. The naïve method is to utilize the relationship inference model [16–18] to infer the hierarchical relationship after being two entity descriptions. However, the result may contain erroneous relationships, since these methods utilize statistical inference models, which often require significant amounts of task-specific training data. Such erroneous relationships have a negative impact on the matching results. It suggests that the hierarchical structure analyzer must be error-free.

In knowledge graphs, low-level entities often serve dual roles, functioning both as independent entities and as attribute values within other entities. Following the insight, we propose a hierarchical structure analyzer (HSA), which takes a knowledge graph with complex entities, denoted $KG = (E, R, A, V)$, as input. It outputs a set of hierarchical relationships R_r^{HS} and a collection of entity sets of different layers, denoted E^0, E^1,..., E^{M}, and the maximum number of layers M. For simplicity, E^m represents the set of entities in the m-th layer. E and E^m satisfy the formula shown in Eq (3.1). In particular, E^0 represents the set of entities, the attribute value of which is not entities of other categories. If the entities (e.g., $e_i^m \in E^m$ and $m \geq 0$) are composite entities, they are preferably dealt with after the low-level entities (e.g., $e_i \in E^{m-1}$), in order to leverage the matching information interactions. The key difference between R_r and R_r^{HS} lies in the semantics of the relational triples. Each relational triple (e.g., $(h_{\mathrm{i}}^{\mathrm{H}}, r^{\mathrm{HS}}, t_j^{\mathrm{L}}) \in R_r^{\mathrm{HS}}$) captures a hierarchical relationship r^{HS} between entities of different categories, where h_i^{H} and t_j^{L} belong to the high-level entity and the low-level entity, respectively.

$$E = \bigcup_{0 \leq m \leq \mathrm{M}} E^m. \tag{3.1}$$

The pseudo-code for the hierarchical structure analyzer procedure is presented in Algorithm 1. The HSA algorithm consists of two steps. 1) *Identification of the hierarchical relationships*: The HSA identifies hierarchical relationships between entities belonging to distinct categories according to the characteristics of composite entities (Lines 1–18). 2) *Hierarchical structure construction*: The HSA constructs a bottom-up hierarchical structure by analyzing the hierarchical relationships between entities. The HSA first identifies entities that occur only as head entities, and designates them as E^0 (Lines 19–23). Then, in each iteration, the HSA assigns all tail entities connected to the head entities at the m-th layer to the $m + 1$-th level. This process continues until all entities are assigned a level (Lines 24–34). Finally, the level indices of entities are inverted—an entity at the m-th layer is reassigned to the (M-m)-th layer (Lines 35–37). As a result, a complete hierarchical structure of entities is constructed.

We construct the hierarchical structures of KG_1 and KG_2 by applying the hierarchical structure analysis procedure to each of them.

3.2. Blocker filter

Existing blocker filters primarily rely on other entity features, such as entities' names and attributes. During the process of filtering unmatched entities, these types of blockers tend to disrupt the structural semantics of candidate entities, resulting in the loss of structural features. However, utilizing a blocker based on dual-layer structural semantics can effectively avoid this issue.

Algorithm 1 : The hierarchical structure analyzer procedure

Input: $KG = (E, R, A, V)$: The i-th knowledge graph with complex entities, where E is the entity set, $R = R_r \cup R_a$ is the relation set, A is the attribute set, and V is the attribute value set.

Output: R_r^{HS}: the set of hierarchical relationships of KG; $\left\{E^0, E^1, ..., E^{\mathrm{M}}\right\}$: a set of different layer entities, where E^m: the set of the complex entity at the m-th layer of KG, M: the maximum number of layers.

Initial mapEntity2AV = $\emptyset$, mapAV2Entity = $\emptyset$, $E^{head} = \emptyset$, $E^{tail} = \emptyset$, $E^{reamain} = E$
Step 1: Hierarchical structure Search phase
for each entity e_i in E of KG **do**
  for each attribute value v_i in V **do**
    if $getName(e_i)$ equals $getName(v_i)$ **then**
      mapEntity2AV.put(e_i,v_i), mapAV2Entity.put(v_i,e_i)
    end if
  end for
  for each attribute triple r_a in R_a **do**
    currentEntity = getEntity(r_a), currentAV = getAttributeValue(r_a)
    if mapAV2Entity.containsKey(currentAV) **then**
      $h_{\mathrm{r}}^{\mathrm{H}}$ = currentEntity, $t_{\mathrm{r}}^{\mathrm{L}}$ = mapAV2Entity.get(currentAV)
      $r_i^{\mathrm{HS}} = (h_{\mathrm{r}}^{\mathrm{H}}, r^{HS}, t_r^{\mathrm{L}})$
      $R_r^{\mathrm{HS}} = R_r^{\mathrm{HS}} \cup \{r_i^{\mathrm{HS}}\}$
    end if
    $R_r = R_r \cup R_r^{\mathrm{HS}}$
  end for
end for
Step 2: Identify the number of layers in the hierarchical structure where the entity is located
for each relational triple r_i in R_r **do**
  $E^{head} = E^{head} \cup \{\text{headEntity}(\mathrm{r}_i)\}$, $E^{tail} = E^{tail} \cup \{\text{tailEntity}(\mathrm{r}_i)\}$
  $E^0 = E \backslash E^{tail}$
  $m = 0$
  while $E^{remain} \neq \emptyset$ **do**
    $E^{reamain} = E - E^0$
    for each $e_i^m \in E^m$ in R_r **do**
      if $e_i^m = h_r^H$ **then**
        $e_i^{m+1} = t_r^L$
        $E^{m+1} = E^{m+1} \cup \left\{e_i^{m+1}\right\}$
      end if
    end for
    $E^{remain} = E^{remain} - E^{m+1}$
    $m = m + 1$
  end while
  $\mathrm{M} = m$
  $E^m = E^{\mathrm{M}-m}$
end for
return R_r^{HS}, $\left\{E^0, E^1, ..., E^{\mathrm{M}}\right\}$, and M

We propose a dual-layer structural feature-based blocker filter for generating candidate entity sets. The blocker effectively utilizes the structural semantic features of entities to rapidly narrow down the scope of entity matching candidate sets while maximizing the retention of potentially aligned entities. Even in scenarios with limited entity structural information, the blocker can effectively exclude unmatched entities and interfering entities, thereby reducing the size of candidate sets and improving the model's execution efficiency. The dual-layer structural feature-based blocker filter first quantifies the structural embedding features of entities. Then the blocker filter constructs candidate sets from two directions, denoted as $KG_1 \rightarrow KG_2$ and $KG_2 \rightarrow KG_1$. In the direction $KG_1 \rightarrow KG_2$, for any entity e_{i1} in KG_1, the top K entities e_{j2} with the highest similarity in KG_2 are selected as the candidate entity set C_{i1}^{K} for entity e_{i1}. The similarity $Sim\left(e_{i1}, e_{j2}\right)$ between any two entities $e_{i1} \in KG_1$ and $e_{j2} \in KG_2$ is calculated using the L_2-norm computation as shown in Eq (3.2). The procedure for the direction $KG_2 \rightarrow KG_1$ is similar to that of $KG_1 \rightarrow KG_2$ and is therefore omitted for brevity.

Next, we take e_{i1} as an example to illustrate the procedure for the construction of the candidate entity set C_{i1}^{K}. First, for any entity $e_{i1} \in KG_1$, we calculate the similarity of the structural features between e_{i1} and all entities e_{j2} in KG_2 using the L_2-norm computation shown in Eq (3.2). Then all similarity scores e_{j2} for entity e_{i1} are arranged in descending order. Finally, the top K entities with the highest similarity scores are selected as the candidate entity set C_{i1}^{K} for entity e_{i1}. Correspondingly, the same method is used to generate the candidate set C_{j2}^{K} of e_{j2}.

$$Sim\left(e_{i1}, e_{j2}\right) = \left\| h\left(e_{i1}\right) - h\left(e_{j2}\right) \right\|_2. \tag{3.2}$$

3.3. Multi-perspective semantic embedding strategy

The existing entity alignment methods are largely constrained by their reliance on single-perspective features, typically based on either structural information or attribute-level textual descriptions. This limitation significantly hinders the ability to capture comprehensive entity semantics, especially in domain-specific knowledge graphs where information is often sparse or incomplete. Furthermore, these methods often neglect the potential value of matching information between different entity layers, resulting in suboptimal alignment performance. To address these limitations, we propose a multi-perspective semantic embedding strategy that systematically integrates structural information, attribute information, corroborative information, and the intrinsic correlations among the aforementioned information. The proposed multi-perspective semantic embedding strategy can fully utilize the matching information flow over low-level entities and high-level entities, in order to achieve more accurate and robust entity alignment.

A distinctive feature of MiAD is the hierarchical matching information enhancement mechanism. For atomic entities, the multi-perspective semantic embedding strategy iteratively refines their representations through matching propagation within the same level. For composite entities, the multi-perspective semantic embedding strategy leverages both intra-level and cross-level interaction mechanisms, where the matching results of low-level entities serve as additional semantic features for their corresponding high-level entities through a recursive enhancement process that significantly improves the quality of entity descriptions and alignment accuracy.

The multi-perspective semantic embedding strategy systematically categorizes domain knowledge graph embeddings into four types: structural embeddings, attribute embeddings, corroborative information embeddings, and joint embeddings. According to this classification, it constructs the corresponding structural semantic aggregator, attribute semantic aggregator, corroborative aggregator, and joint aggregator. Specifically, the multi-perspective semantic embedding strategy consists of four core components: (i) The structural semantic aggregator that employs a dual-layer GCN and a self-attention mechanism to capture both direct and indirect structural relationships; (ii) The attribute semantic aggregator that combines low-level and high-level attribute embeddings through advanced fusion strategies; (iii) The corroborative aggregator that integrates position-aware attention mechanisms with pretrained BERT models to deeply model the external textual semantic information; and (iv) The joint aggregator that dynamically integrates multi-perspective feature representations using an adaptive weighting mechanism, fully utilizing the matching information flow between different hierarchical levels.

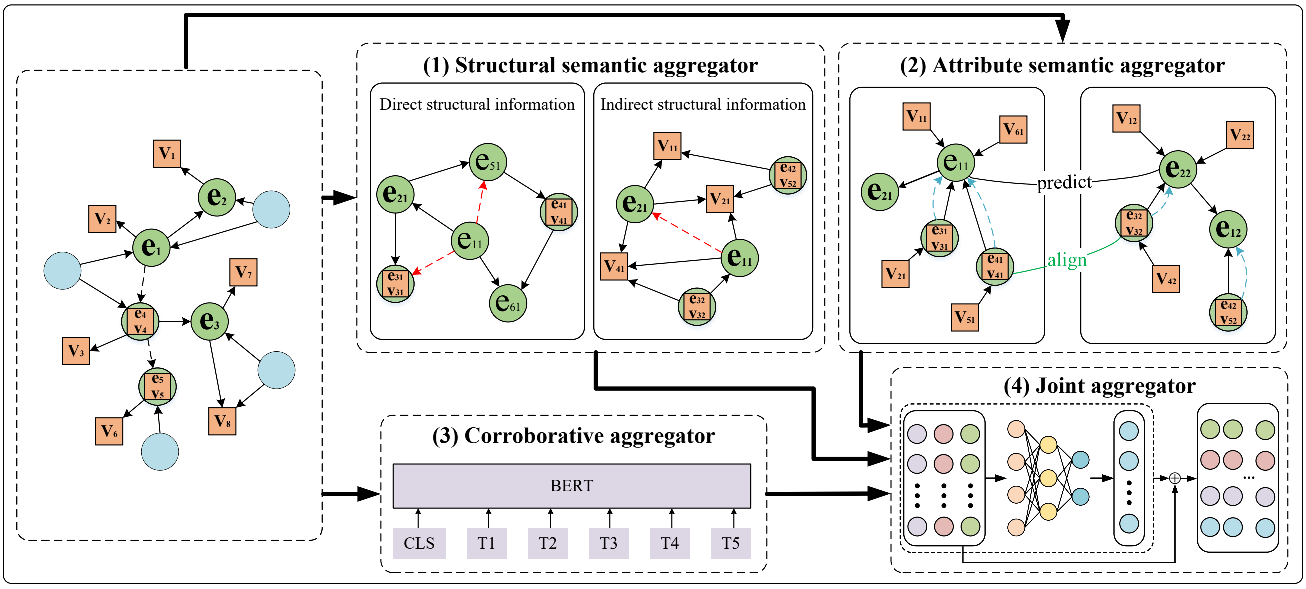

Figure 4. The architecture of the multi-perspective semantic embedding strategy.

3.3.1. Structural semantic aggregator

In real-world scenarios, the abundance of long-tail entities leads to sparsely connected embedding spaces, which subsequently impair the overall alignment performance. Moreover, conventional structural semantic feature constructors fail to exploit the high-confidence matching information derived from low-level entities. In addition, neighboring entities vary in their importance and contribution to target entities. However, manually assigning weights to each neighbor is both impractical and labor-intensive. Consequently, there is a critical need for an adaptive weighting mechanism that can automatically estimate the importance of neighboring entities.

To address the aforementioned issues, we propose a structural semantic aggregator that constructs entity representations using dual-layer structural information. It offers three main advantages: (i) It captures entity features from both direct and indirect structural perspectives, effectively integrating two types of features into comprehensive dual-layer structural embeddings. (ii) High-confidence matching information from low-level entities can serve as shared attribute values for the corresponding high-level entities, thereby enabling the capture of richer structural and semantic information. (iii) To account for the varying importance of neighboring entities, a self-attention mechanism is introduced to dynamically evaluate their relevance and assign attention weights accordingly.

The procedure by which the structural semantic aggregator captures both direct and indirect structural information is described as follows. For modeling the direct structural information, a GCN is used to perform multi-round feature learning and information propagation on same-layer structural data. For modeling the indirect structural information, high-confidence bidirectional matching among low-level entities reveals the latent semantic hints of high-level entities. To address the sparsity of relationships involving long-tail entities and to fully leverage matching information interactions, a structural semantic aggregator integrates two types of structural associations during the initialization of entity features: (i) Semantic associations between long-tail entities and other entities based on shared attribute characteristics; and (ii) cross-level mappings that facilitate the generation of indirect structural embeddings.

We construct a structural semantic aggregator based on dual-layer structural information. A detailed derivation of the formulas associated with the structural semantic aggregator is provided below.

Next, we take e_i as the example to illustrate how to quantify the structural semantic embedding $h_i^{(l)}$. Specifically, $h_i^{(l)}$ is computed by applying the function $f(\cdot)$ to the aggregated representation $h_i^{(l-1)}$ from the previous hidden layer, as shown in Eq (3.3). Here, the notation i' denotes an index corresponding to a neighboring entity of e_i.

$$h_i^{(l)} = f\left(h_i^{(l-1)}, M, W^{(l-1)}\right). \tag{3.3}$$

The mathematical expression of $f(\cdot)$ is shown in Eq (3.4), where $b^{(l-1)}$ indicates a bias, which can be automatically adjusted for updates by backpropagation. The activation function σ is implemented as the *LeakyReLu* function.

$$h_i^{(l)} = \sigma\left(Mh_i^{(l-1)}W^{(l-1)} + b^{(l-1)}\right). \tag{3.4}$$

1) Calculating the adjacency matrix M

In the domain knowledge graph, the limited connectivity among entities leads to the sparsity of the adjacency matrix. To address this issue, the dual-layer adjacency matrix M is reconstructed by incorporating both the direct structural information and the indirect structural information. The formulation of M is provided in Eq (3.5).

$$M = M^d + M^i. \tag{3.5}$$

In Eq (3.5), M^d represents the adjacency matrix derived from the direct structure information between entities, whereas M^i represents the adjacency matrix constructed from the indirect structure information, inferred on the basis of the attribute values shared among entities.

The element $M^d_{ii'} \in M^d$ indicates whether there is a direct correlation between entity e_i and entity $e_{i'}$, and its value is obtained according to Eq (3.6).

$$M^d_{ii'} = \begin{cases} 1, & \text{there is a relationship between } e_i \text{ and } e_{i'}. \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

The sparsity of inter-entity relationships leads to a large number of zero entries in M^d. To alleviate this issue, M^i is designed to capture indirect relationships among entities that share common attribute values. $M^i_{ii'} \in M^i$ is assigned a small initial value k to reduce the influence of excessive indirect neighbors on the process of structural feature aggregation, as shown in Eq (3.7).

$$M^i_{ii'} = \begin{cases} k, & \text{if } e_i \text{ and } e_{i'} \text{ share at least one common attribute} \\ 0, & \text{otherwise} \end{cases}, \quad \text{where } 0 < k \leq 1. \tag{3.7}$$

2) Calculating the weight matrix $W^{(l-1)}$

For a given entity, its neighboring entities may contribute with varying degrees of importance. To this end, we reconstruct the weight matrix W of the GCN by combining the characteristics of entities. Next, we show how to calculate the weight matrix of the $l-1$-th hidden layer of the GCN, denoted $W^{(l-1)}$. $W^{(l-1)}$ is a linear combination of the self-learning weight of the GCN (denoted $W_g^{(l-1)}$) and the self-attention weight learned by the self-attention mechanism (denoted $W_a^{(l-1)}$), as shown in Eq (3.8). $W_g^{(l-1)}$ enables the embedding process to model the neighboring structure information effectively, while the self-attention weight $W_a^{(l-1)}$ focuses more on the importance of the relationship between the current entity and its neighbors.

$$W^{(l-1)} = \alpha \cdot W_g^{(l-1)} + (1-\alpha) \cdot W_a^{(l-1)}. \tag{3.8}$$

The procedure for solving $W_a^{(l-1)}$ is described below.

(i) The attention coefficient $e_{ii'}$, which quantifies the relevance of the neighboring entity $e_{i'}$ to the target entity e_i, is calculated by Eq (3.9). The specific calculation process is as follows: (a) The eigenvectors h_i and $h_{i'}$, associated with the entities e_i and $e_{i'}$, are transformed via a self-learned weight vector $\boldsymbol{\theta}$. The transformed representations (i.e., θh_i and $\theta h_{i'}$) are then concatenated to construct enhanced high-dimensional feature vectors that capture richer semantic information. (b) The attention weight vector a^T is employed to adaptively assign weights to different features. (c) $e_{ii'}$ is computed after transformed by the nonlinear activation function *LeakyReLu* and it indicates the importance of the neighboring entity $e_{i'}$ among all neighboring entities of the entity e_i

$$e_{ii'} = LeakyReLu(a^T[\boldsymbol{\theta} h_i \| \boldsymbol{\theta} h_{i'}]). \tag{3.9}$$

(ii) In order to facilitate the optimization of $\boldsymbol{\theta}$, the softmax normalization function is applied to the attention coefficient $e_{ii'}$, and the corresponding attention score $a_{ii'}$ is derived according to Eq (3.10)

$$a_{ii'} = softmax_{i'}(e_{ii'}) = \frac{\exp(e_{ii'})}{\sum_{k \epsilon N_i} \exp(e_{ik})}, \ i', k \epsilon N_i. \tag{3.10}$$

(iii) $W_a^{(l-1)}(i, i')$, the self-attention weight of the neighboring entity $e_{i'}$ with respect to the target entity e_i, is computed according to Eq (3.11). $W_a^{(l-1)}$ is derived after obtaining all the self-attention weights of entities at the layer $(l-1)$.

$$W_a^{(l-1)}(i, i') = \begin{cases} a_{ii'}, & \text{there is an attention coefficient between } e_i \text{ and } e_{i'}. \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

3) Calculating the adjacency matrix $\widehat{M}$

In order to better capture the bidirectional relationship between entities, the adjacency matrix M of a directed graph, shown in Eq (3.5), is converted into an undirected graph by symmetric normalization. At the same time, in order to make full use of the entity feature learning process, the self-joining relation I is added to the adjacency matrix M. After symmetric normalization and self-concatenation, the resulting adjacency matrix $\widehat{M}$ is mathematically defined in Eq (3.12).

$$\widehat{M} = D^{-\frac{1}{2}} M D^{-\frac{1}{2}} + I. \tag{3.12}$$

On the basis of Eq (3.12), the degree matrix $\widehat{D}$ is computed from $\widehat{M}$. The calculation of each element $\widehat{D}_{ii'} \in \widehat{D}$ is defined in Eq (3.13). Subsequently, $\widehat{D}$ is diagonalized as shown in Eq (3.14) to obtain $\widehat{D}^{-\frac{1}{2}}$.

$$\widehat{D}_{ii} = \sum\nolimits_{i'} \widehat{M}_{ii'}. \tag{3.13}$$

$$\widehat{\boldsymbol{D}}^{-\frac{1}{2}} = diag(\frac{1}{\sqrt{\widehat{\boldsymbol{D}}_{11}}}, \frac{1}{\sqrt{\widehat{\boldsymbol{D}}_{22}}}, ..., \frac{1}{\sqrt{\widehat{\boldsymbol{D}}_{nn}}}). \tag{3.14}$$

After l iterations of the aforementioned aggregation process, the final structural representation $h_i^{(l)}$ of entity e_i at the l-th hidden layer is derived as shown in Eq (3.15). The final output from the final iteration is adopted as the structural feature representation h_i^s of the entity e_i.

$$h_i^{(l)} = \sigma\left(\widehat{\boldsymbol{D}}^{-\frac{1}{2}}(D^{-\frac{1}{2}}(M^d + M^i)D^{-\frac{1}{2}} + I)\widehat{\boldsymbol{D}}^{-\frac{1}{2}} h_i^{(l-1)}(\alpha \cdot \boldsymbol{W}_g^{(l-1)} + (1-\alpha) \cdot \boldsymbol{W}_a^{(l-1)}) + b^{(l-1)}\right). \tag{3.15}$$

3.3.2. Attribute semantic aggregator

Most existing methods only embed attribute names in knowledge graphs, leading to limited embedded features and loss of semantic information. In contrast, attribute values possess higher-dimensional and more granular characteristics, enabling the extraction of richer entity features. Entity attribute information can be divided into two levels—direct and indirect—depending on whether the values originate from the entities themselves or are inferred through matching. Direct attribute information refers to the original attribute–value pairs that are explicitly linked to the entities. Indirect attribute information arises from low-level entity alignment, where the aligned composite entities serve as shared attributes of their corresponding high-level entities. Both direct and indirect attribute information enrich the attribute descriptions of entities and enhance the completeness of their semantic representations. In addition, because both the entities' names and attribute values are textual and originate from the entities themselves, and the entities' names also play an important role in entity alignment, we incorporate them jointly into the procedure of the semantic attribute embeddings of entities.

To comprehensively capture the semantic information embedded in both entity names and attribute values, we propose an attribute semantic aggregator composed of two key components: A word embedding module and a tuple embedding module. The attribute semantic aggregator simultaneously takes direct and indirect attribute information, along with the entities' names, as input, and extracts semantic features from both the entities' names and attribute values. As shown in Figure 5, first, the entities' names and attribute values are encoded into semantic vectors using a word embedding module. Then the tuple embedding module computes the term frequency and importance weights of the attribute terms, which are then used to perform a weighted average over their semantic vectors. Subsequently, singular value decomposition (SVD) is applied for dimensionality reduction and redundancy elimination. Finally, the rectified linear unit (ReLU) activation function is applied to introduce nonlinearity into the transformed feature space, thereby producing attribute representation vectors.

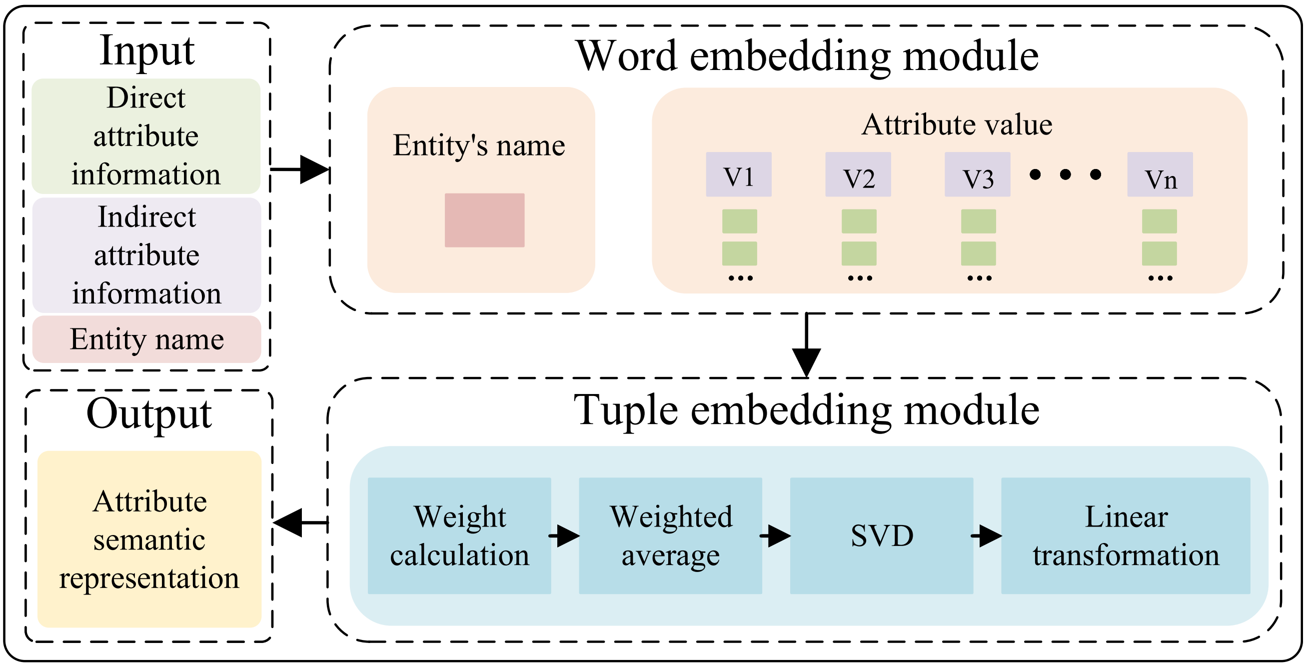

Figure 5. Attribute semantic aggregator.

1) Word embedding module

Domain knowledge graphs typically encompass a large number of domain-specific terms, many of which are out-of-vocabulary (OOV) words. Traditional word embedding methods like Word2Vec [19] cannot generate representations for these unseen words. To address this issue, we employ fastText in the word embedding module. FastText [20] can generate meaningful embeddings for OOV words by decomposing words into character n-grams and offer robustness against spelling variations and domain-specific terminology. The input consists of direct attribute values, indirect attribute values derived from aligned entities, and entity names.

We take entity e_i as an illustrative example to describe the processing pipeline of the word embedding module. (i) For each entity e_i, its entity name e_i^z and the associated attribute values $V = \left[v_i^1, v_i^2, \ldots, v_i^n\right]$ are concatenated to form a training sequence $e_i^s = \left[e_i^z, v_i^1, v_i^2, \ldots, v_i^n\right]$. Each word in the sequence is decomposed into a sequence of characters. In addition, $t_i = \left[word_i^1, word_i^2, \ldots, word_i^{\mathrm{T}}\right]$ is obtained by performing word segmentation on e_i^s, where $word_i^k$ denotes the k-th word of entity e_i. (ii) For each $word_i^k$ in t_i, using Wikipedia as an external corpus, fastText generates a corresponding character-level embedding vector $\mathbf{e}_i^k$, resulting in the character-level embedding sequence $h_i^w = \left[\mathbf{e}_i^1, \mathbf{e}_i^2, \ldots, \mathbf{e}_i^{\mathrm{T}}\right]$. (iii) Finally, t_i and h_i^w, derived from e_i^s, are fed into the tuple embedding module to further capture the semantic associations among the entity's attributes.

2) Tuple embedding module

These vectors generated from the word embedding module are independent and fail to capture the contextual relationships within attribute sequences. Moreover, entities have varying numbers of attributes, resulting in attribute representations with inconsistent dimensions. To address these issues, we employ a tuple embedding module that aggregates word vectors into fixed-dimensional entity representations while preserving the semantic relationships among attributes. We utilize SIF [21] to generate unified attribute representations through the following steps.

(i) For each $word_i^k$ in t_i, we first compute its word frequency f_i^k, then calculate the weight for $word_i^k$ using the smoothed sublinear term frequency as shown in Eq (3.16).

$$w_i^k = \frac{a}{a + f_k^i}. \tag{3.16}$$

Here, a is a smoothing parameter that down-weights the f_i^k of frequent but potentially less informative words (e.g., function words like "is", "the"). This prevents common words from dominating the attribute representation.

(ii) To ensure consistent embedding dimensionality, the vector representation of each entity e_i, denoted v_i, is computed as the weighted average of its character-level embeddings, as defined in Eq (3.17). Specifically, each t_i contains $|t_i|$ character-level word vectors. This ensures consistent dimensionality while preserving the semantic relevance of the attribute terms.

$$v_i = \frac{1}{\sum_{k=1}^{|t_i|} w_i^k} \sum_{k=1}^{|t_i|} w_i^k \cdot \mathbf{e}_i^k. \tag{3.17}$$

(iii) To eliminate biases from common, nondiscriminative semantics introduced by the weighted averaging operation, we perform SVD to derive the final embedding v_i', as shown in Eq (3.18).

$$v_i' = v_i - uu^T v_i. \tag{3.18}$$

(iv) Through the abovementioned tuple embedding process, we obtain the attribute representation v_i' of e_i. To derive the final attribute feature representation, we apply a linear transformation followed by a nonlinear activation, where W_a and b_a are learnable parameters, as shown in Eq (3.19).

$$h_i^{\mathrm{a}} = \mathrm{ReLU}(W_a v_i' + b_a). \tag{3.19}$$

3.3.3. Corroborative aggregator

In addition to internal information such as the structure and attributes, domain knowledge graphs also contain abundant external information (such as entity descriptions [22], entity images [23], etc.). Existing domain knowledge graphs commonly face the problem of insufficient internal features, which makes it difficult for state-of-the-art entity alignment (EA) solutions based on representation learning to produce ideal matching results [24].

To address the issue of insufficient intrinsic features in domain knowledge graphs, we propose a BERT-based corroborative aggregator which can effectively extract the external information of entities to compensate for the limitations of the internal features. Our proposed corroborative aggregator takes the external information, such as entity descriptions, as input and employs a BERT [25]) model composed of multiple transformer layers to capture corroborative information. It effectively extracts supplementary entity characteristics and enhances the overall quality of entity feature representations.

The aggregator first leverages the WordPiece algorithm to tokenize the entity descriptions into discrete sequences. Then fastText is leveraged to produce the wordpiece embedding c_i^{word}, the position embedding c_i^{pos}, and the weight embedding c_i^{weight} of the i-th token, as defined in Eq (3.20). Weight embeddings are calculated using the same method for solving character weights as shown in Eq (3.16). The wordpiece embedding, the position embedding, and the weight embedding of each token are added together to construct the embedding layer of BERT. Finally, the BERT model is employed to perform corroborative information feature learning by processing $h_i^{(l-1)}$ through a series of multi-head transformer layers, as defined in Eq (3.21). Next, $h_i^{(0)} = [c_1^{(0)}, ...c_N^{(0)}]$, the initialized embedding sequence, is fed into the transformer layers.

$$c_i^{(0)} = c_i^{word} + c_i^{pos} + c_i^{weight}(i = 1, ..., N). \tag{3.20}$$

$$h_i^{(l)} = Transformer(h_i^{(l-1)}). \tag{3.21}$$

3.3.4. Joint aggregator

Directly leveraging the aforementioned embedding features entails several limitations. First, the embedding features from the three perspectives are independently learned from separate aggregation processes, which results in the loss of intrinsic correlations among them. However, since all three embeddings describe the same entity, it is essential to uncover their inherent semantic relationships. Second, the use of distinct aggregators for different perspectives often leads to inconsistencies in embedding dimensionality, thereby necessitating a unified representation to enable the effective integration of features from all three perspectives. Therefore, we propose the joint aggregator to capture the correlations among the different perspectives in the form of dense vector representations.

To address the weak intrinsic correlation among the three-perspective embedding features, the joint aggregator employs a multi-layer perceptron (MLP) to explicitly model the relationships among the structural features h_i^{s}, the attribute features h_i^{a}, and the corroborative features h_i^{c}. This yields a unified internal representation of the entity, as shown in Eq (3.22)

$$h_i^{\mathrm{j}} = MLP\left(h_i^{\mathrm{s}} + h_i^{\mathrm{a}} + h_i^{\mathrm{c}}\right). \tag{3.22}$$

We obtain the comprehensive embedding representation of e_i, denoted h_i, via the concatenation operation, which integrates the structural features, attribute features, corroborative features, and intrinsic correlation features, and can fully retain the embedding features from different perspectives, as shown in Eq (3.23)

$$h_i = Contact\left(h_i^{\mathrm{s}} + h_i^{\mathrm{a}} + h_i^{\mathrm{c}} + h_i^{\mathrm{j}}\right). \tag{3.23}$$

3.4. Spatial mapping and heterogeneous entity relationship inference

3.4.1. Spatial mapping

To address the potential inconsistency of embedding spaces between knowledge graphs from different sources, a space mapping mechanism is required to unify the entity spaces. The spatial mapping mechanism takes the comprehensive embeddings of entities from KG_1 and KG_2 as input, and produces the unified comprehensive embeddings of the corresponding entities as output by applying the transformation matrix T as defined in Eq (3.24).

Next, we explain how to calculate T. Specifically, some aligned entities were manually preselected and processed through the multiple perspective semantic encoder along with other entities to generate embedding features. The space mapping mechanism solves for the transformation matrix T via Eq (3.24) that can be utilized to unify embedding spaces for knowledge graphs from different domains. Here, $h^n_{G_1}$ and $h^n_{G_2}$ represent the embedding feature vectors of the n-th pair of matched entities between KG_1 and KG_2.

$$\begin{cases} h^1_{G_1} = Th^1_{G_2} \\ h^2_{G_1} = Th^2_{G_2} \\ \quad\vdots \\ h^n_{G_1} = Th^n_{G_2} \end{cases} . \tag{3.24}$$

3.4.2. Heterogeneous entity relationship inference

The heterogeneous entity relationship inference module, a key component of MiAD, addresses the challenge of identifying equivalent entities in knowledge graphs that exhibit structural and semantic diversity. This module takes the unified comprehensive embedding representations of entities as input and outputs binary classification results indicating whether the entity pairs are aligned or not, along with confidence scores for these alignments. By formulating entity alignment as a binary classification task, the inference module leverages the MLP to map the unified comprehensive embedding representations of the entities to the probabilities of the equivalent relationship among the heterogeneous entity pairs through multiple layers of nonlinear transformations.

Since the more two entities match, the greater their semantic similarity and structural similarity, we refined the loss function of the MLP in order to enhance the generalization of entity alignment. The loss function $\mathcal{L}_S$ of the inference model is designed to minimize the differences in the aligned pairs $\left(\mathrm{e}_{i1}, \mathrm{e}_{j2}\right)$, as shown in Eq (3.25), where $e_{i1} \epsilon E_1$ and $e_{j2} \epsilon C^K_{i1}$ represent the entity pairs that are judged to be matched, N_{i1} and N_{j2} represent the entity sets formed by the neighboring entities. The semantic similarity between e_{i1} and e_{j2}, denoted $Sim\left(e_{i1}, e_{j2}\right)$, is computed according to Eq (3.2), where $h(e_{i1})$ and $h(e_{j2})$, respectively, represent the joint embedding representation; the L_2 norm is used to calculate the similarity between $h(e_{i1})$ and $h(e_{j2})$. $\mathcal{L}_S$ ensures that the aligned entities not only possess similar semantic representations but also maintain consistent structural relationships with their neighboring entities, thereby preserving both local and global semantic coherence across knowledge graphs.

$$\mathcal{L}_S = \sum_{e_{i1} \in KG_1} \left\| \frac{1}{|N_{i1}|} \sum_{e_{i'1} \in N_{i1}} Sim\left(e_{i1}, e_{i'1}\right) - \frac{1}{|N_{j2}|} \sum_{e_{j'2} \in N_{j2}} Sim\left(e_{j2}, e_{j'2}\right) \right\| . \tag{3.25}$$

The procedure of the heterogeneous entity relationship inference module is as follows. (i) Neighborhood construction: For each entity (e.g., e_i) in the source knowledge graphs, the module constructs the set of neighboring entities set N_i by collecting entities that have direct relational connections to e_i. Similarly, for each entity e_j in the target knowledge graph, the module constructs the neighboring entity set of e_j, denoted N_j. (ii) Similarity computation: The module computes the similarity scores of entity pairs between each entity and its neighboring entity, according to Eq (3.2). (iii) Loss optimization: The module applies the relational constraint loss function to optimize the alignment predictions through Eq (3.25). (iv) Alignment prediction: The MLP-based classifier is trained using the loss function. After training, the module outputs a set of aligned pairs whose similarity scores fall below a predefined threshold. Through the heterogeneous entity relationship inference module, equivalent entities can be effectively identified across diverse knowledge graphs, despite variations in their structure and semantics.

3.5. Iterative optimization strategy

We observe that facilitating the interaction of matching information can significantly enhance the performance of entity alignment. Matching information interactions can be classified into two categories: intra-level interactions and cross-level interactions. We propose a novel iterative optimization strategy that jointly exploits intra-level and cross-level interactions to refine both heterogeneous entity relationship inference and the multi-perspective semantic encoders, as illustrated in Figure 6.

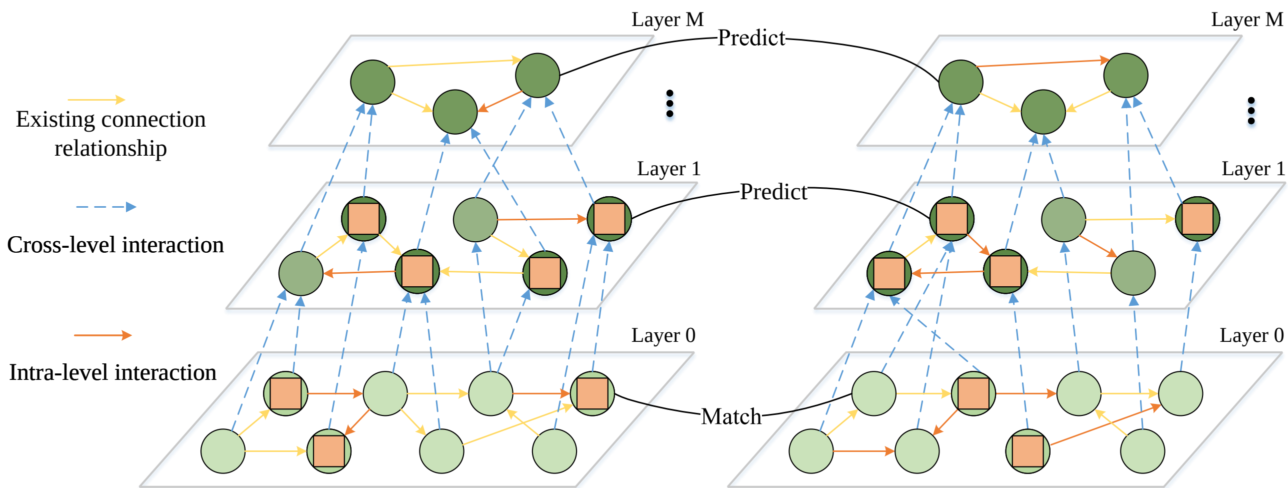

Figure 6. A schematic illustration of the iterative optimization strategy.

A schematic diagram of the iterative optimization strategy is presented in Figure 6. Assuming that the entities in the heterogeneous knowledge graphs with composite entities can be divided into M layers. Intra-level interactions identify high-confidence aligned entity pairs from the outputs of the relationship inference module and use them to infer new relational triples (orange lines), thereby enriching the knowledge graphs. This process iterates until no further performance improvement is achieved. Cross-level interactions (blue dashed lines) utilize the equivalence among low-level entities to enhance the semantic embeddings of high-level entities. Existing relational triples are denoted by yellow lines.

The intra-level interactions are utilized to identify the high-confidence aligned entity pairs from the predicted matching pairs derived from the entity relationship inference module and to infer the newly relational triples, which are highlighted in orange, thereby enriching the knowledge graphs. Once the high-confidence aligned entity pairs can not enhance the performance of the iterative entity relationship inference model, the mutual reinforcement between the entity relationship inference module and the intra-level interaction mechanism is terminated. As illustrated by the blue dashed lines, cross-level interactions leverage the equivalence relationships among low-level entities to enhance the multi-perspective semantic embeddings of high-level entities. The yellow lines illustrate the existing relational triples connecting entities in the knowledge graphs.

In short, the iterative optimization strategy applies layer-wise optimization to heterogeneous entity relationship inference and incrementally enhances the embedding representations of high-level entities by analyzing the matching information of low-level entities.

3.5.1. The intra-level interaction mechanism

The intra-level interaction occurs among entities at the same level and can effectively improve the matching results. The following illustrative example motivates the intra-level interaction mechanism.

Assume that a certain entity $e_{i1}^{m} \in KG_1$ located in the m-th layer of the matching information hierarchy, has a direct connection with its neighboring entity $e_{i'1}^{m} \in KG_1$ according to the relational triples of KG_1, whereas their counterparts $e_{j2}^{m} \in KG_2$ and $e_{j'2}^{m} \in KG_2$ do not share such a direct connection in KG_2. Given that the entity pair $p_{i1,j2} = \{(e_{i1}^{m}, e_{j2}^{m}) | e_{i1}^{m} \in KG_1, e_{j2}^{m} \in KG_2\}$ corresponds to the same real-world entity; similarly, $p_{i'1,j'2} = \{(e_{i'1}^{m}, e_{j'2}^{m}) | e_{i'1}^{m} \in KG_1, e_{j'2}^{m} \in KG_2\}$ also denotes an identical real-world entity. It can be inferred that KG_2 lacks a direct connection between e_{j2}^{m} and $e_{j'2}^{m}$. This structural inconsistency may impair the alignment performance. To address this issue, the intra-level interaction mechanism aims to enrich the missing structural information by leveraging the predicted results generated by the inference module, as detailed in Section 4.4.2.

A central challenge is ensuring that the intra-level interaction mechanism can confidently identify entity pairs that refer to the same real-world entity. It is observed that a higher matching probability derived from the inference module indicates greater confidence in the aligned entity pairs. The intra-level interaction mechanism employs a semantic similarity-based filter to refine the predicted aligned entity pairs, with the goal of constructing a high-confidence set of aligned entity pairs while reducing false positive errors. Here, the high-confidence set consists of high-confidence aligned entity pairs that refer to the same real-world entity. False positive errors refer to truly unmatched entity pairs that are incorrectly predicted as aligned.

The semantic similarity-based filter first initializes the entity pairs of the seed as the high-confidence set, then identifies all high-confidence entity pairs from the predicted aligned pairs generated by the inference model and subsequently adds them to the high-confidence set.

A predicted aligned entity pair at the m-th level (e.g., $p_{i1,j2} = \{(e_{i1}^{m}, e_{j2}^{m}) \mid e_{i1}^{m} \in KG_1, e_{j2}^{m} \in KG_2\}$) is considered high-confidence if it satisfies the mutual maximum similarity constraint. In other words, the alignment (e_{i1}^{m}, e_{j2}^{m}) is regarded as high-confidence only when each entity is the most semantically similar to the other within its respective candidate set. This suggests two conditions that must be satisfied.

(i) For $e^m_{i1} \in KG_1$, the entity $e^m_{j2} \in KG_2$ must yield the highest similarity score among all entities in the top K candidate set derived in the $KG_1 \to KG_2$ direction, as formulated in Eq (3.26).

(ii) Conversely, for $e^m_{j2} \in KG_2$, the corresponding entity $e^m_{i1} \in KG_1$ must also yield the highest similarity score within the top K candidates along the $KG_2 \to KG_1$ direction, as formulated in Eq (3.27).

$$\mathrm{sim}(e^m_{i1}, e^m_{j2}) = \max_{e^m_{j2} \in C^{\mathrm{K}}_{i1}(m)} \mathrm{sim}(e^m_{i1}, e^m_{j2}). \tag{3.26}$$

$$\mathrm{sim}(e^m_{j2}, e^m_{i1}) = \max_{e^m_{i1} \in C^{\mathrm{K}}_{j2}(m)} \mathrm{sim}(e^m_{j2}, e^m_{i1}). \tag{3.27}$$

Here $\mathrm{sim}(\cdot,\cdot)$ refers to the similarity function as formulated in Eq (3.2); $C^{\mathrm{K}}_{i1}(m)$ consists of the top K candidate entities for e^m_{i1}, ranked by similarity scores and retrieved along the $KG_1 \to KG_2$ direction; and $C^{\mathrm{K}}_{j2}(m)$ consists of the top K candidate entities for e^m_{j2}, ranked by similarity scores and retrieved along the $KG_2 \to KG_1$ direction.

1) The mutual reinforcement between the heterogeneous entity relationship inference module and the intra-level interaction mechanism through iterative interactions

Next, we elaborate on how the heterogeneous entity relationship inference module and the intra-level interaction mechanism mutually reinforce each other through iterative interactions to improve the entity alignment performance, as detailed in Algorithm 2 (Lines 2–28). To illustrate this process, we take the entity alignment procedure for the m-th level entities as an example, demonstrating how the entity relationship inference model is iteratively optimized via the intra-level interaction mechanism.

For the entities of the m-th level that exist in E^m, the i-th iteration entity relationship inference model (denoted $M^{\mathrm{infer}}_i(m)$) is trained and outputs a set of predicted matched entity pairs, denoted P^M_i and a set of predicted unmatched entity pairs, denoted P^{UM}_i (Lines 4–5). The intra-level interaction mechanism takes P^M_i as input, then generates a high-confidence set denoted $\overline{P^M_i}$ and $R^{\mathrm{new}}_i(m)$ (Line 16). The iteration process is terminated if the high-confidence aligned entity pairs cannot enhance the performance of the iterative entity relationship inference model (Lines 8–26).

2) Termination conditions of the iterative entity relationship inference model

The iterative entity relationship inference model terminates under two conditions: (i) No new high-confidence entity pairs are identified, as indicated in Lines 8–9; (ii) the performance of the current iteration inference model is superior to that of the next iteration or the performance difference is less than $1e-6$, as indicated in Lines 19–25.

3.5.2. The cross-level interaction mechanism

Cross-level interactions occur among entities at different levels. Through the cross-level interaction mechanism, the matching information from low-level entities can enhance the semantic representations of high-level entities, thereby improving the performance of entity alignment for high-level entities.

Algorithm 2 : The intra-level interaction mechanism procedure

Input: E^m: the entities at the m-th level, $S_{seed}(m)$: the seed set of aligned entity pairs at the m-th level.

Output: $bestM^{\text{infer}}(m)$: the best optimized inference model, $bestP^M(m)$: a set of predicted matched entity pairs derived from $bestM^{\text{infer}}(m)$, $bestP^{UM}(m)$: a set of predicted unmatched entity pairs derived from $bestM^{\text{infer}}(m)$, $bestR^{\text{new}}(m)$: the new relational triples derived from the intra-level interaction mechanism, $best\overline{P^M}(m)$: the high-confidence aligned entity pairs.

1. //The mutual reinforcement between the heterogeneous entity relationship inference module and the intra-level interaction mechanism through iterative interactions on E^m
2. Initialize $i \leftarrow 0$, optimizeFlag $\leftarrow True$, $bestM^{\text{infer}}(m) \leftarrow NULL$, $\overline{P_i^M} \leftarrow$ the matched entity pairs in $S_{seed}(m)$, $currTrainSet(m) \leftarrow S(m)$, $bestR^{\text{new}}(m) \leftarrow \emptyset$, $bestP^M(m) \leftarrow \emptyset$, $bestP^{UM}(m) \leftarrow \emptyset$, $best\overline{P^M}(m) \leftarrow \emptyset$
3. **while** optimizeFlag **do**
4. Train the inference model $M_i^{\text{infer}}(m)$ using $currTrainSet(m)$
5. Obtain the predicted matched pairs P_i^M and unmatched pairs P_i^{UM} by applying the inference model $M_i^{\text{infer}}(m)$ over the candidate entity pairs of E^m
6. Obtain a high-confidence set $\overline{P_i^M}$ and $R_i^{\text{new}}(m)$ by applying the intra-level interaction mechanism over P_i^M and the matched entity pairs in $currTrainSet(m)$ according to Eq (3.27) and Eq (3.26)
7. $bestM^{\text{infer}}(m) \leftarrow M_i^{\text{infer}}(m)$, $bestR^{\text{new}}(m) \leftarrow R_i^{\text{new}}(m)$, $bestP^M(m) \leftarrow P_i^M$, $bestP^{UM}(m) \leftarrow P_i^{UM}$, $best\overline{P^M}(m) \leftarrow \overline{P_i^M}$// update the inference model
8. **if** $\overline{P_i^M} \subseteq currTrainSet(m)$ **then**
9. optimizeFlag = False; **break**;
10. **else**
11. Complete the graph according to $bestR^{\text{new}}(m)$
12. Recompute $H^s(E^m)$ according to Eq (3.15)
13. $i \leftarrow i + 1$; $tempTrainSet(m) \leftarrow currTrainSet(m) \cup \overline{P_i^M}$
14. Obtain the optimized inference model $M_i^{\text{infer}}(m)$ by training on $tempTrainSet(m)$
15. Obtain the predicted matched pairs P_i^M and unmatched pairs P_i^{UM}
16. Obtain $\overline{P_i^M}$ and $R_i^{\text{new}}(m)$ derived from the intra-level interaction mechanism
17. Obtain the loss of $M_i^{\text{infer}}(m)$ over E^m, denoted $Loss(M_i^{\text{infer}}(m))$
18. Obtain the loss of $M_{i-1}^{\text{infer}}(m)$ over E^m, denoted $Loss(M_{i-1}^{\text{infer}}(m))$
19. **if** $(Loss(M_i^{\text{infer}}(m)) \leq Loss(M_{i-1}^{\text{infer}}(m))) \wedge (|Loss(M_i^{\text{infer}}(m)) - Loss(M_{i-1}^{\text{infer}}(m))| \geq (1e-6)$ **then**
20. $currTrainSet(m) \leftarrow tempTrainSet(m)$
21. $bestM^{\text{infer}}(m) \leftarrow M_i^{\text{infer}}(m)$; $bestR^{\text{new}}(m) \leftarrow R_i^{\text{new}}(m)$;$bestP^M(m) \leftarrow P_i^M$, $bestP^{UM}(m) \leftarrow P_i^{UM}$,$best\overline{P^M}(m) \leftarrow \overline{P_i^M}$ // update the inference model
22. **continue**
23. **else**
24. optimizeFlag = False; **break**
25. **end if**
26. **end if**
27. **end while**
28. **return** $bestM^{\text{infer}}(m)$, $bestP^M(m)$, $bestP^{UM}(m)$, $bestR^{\text{new}}(m)$, $best\overline{P^M}(m)$

Since certain attribute values of high-level entities are represented by low-level entities (as illustrated in Figure 6), given a high-confidence set of low-level entities, the equivalence of these attribute values of high-level entities can be established. Such equivalence relationships among low-level entities can be exploited to improve the effectiveness of both structural and attribute semantic encoders for the high-level entities. (i) For the structural semantic encoder, these equivalence relationships among low-level entities can enrich the indirect adjacency matrix M^i of the high-level entities. (ii) For the attribute semantic encoder, these equivalence relationships among low-level entities can be translated to common attribute values of high-level entities, thereby enhancing the attribute semantic embeddings of high-level entities.

Assume that the descriptions of the low-level entities are located in the (m-1)-th layer and that the descriptions of the high-level entities are located in the m-th layer. The procedure of the cross-level interaction mechanism is detailed in the pseudo-code of Algorithm 3.

After the alignment of low-level entities, the corresponding intra-level interaction mechanism and the entity relationship inference module cooperate to produce a high-confidence set (e.g., $best\overline{P^M}(m\text{-}1)$) at the ($m$-1)-th level. For high-level entities, the matching information of low-level entities is represented in $best\overline{P^M}(m\text{-}1)$. This allows us to enrich the attribute triples of high-level entities in Line 1 and the relational triples in Line 2. The cross-level interaction mechanism further enhances the structural and attribute semantic embeddings of the high-level entities (i.e., entities at the m-th level) in Lines 3 and 4, respectively. Especially, if $m = 0$, $best\overline{P^M}(m\text{-}1)$ is a empty set.

Algorithm 3 : The cross-level interaction mechanism procedure

Input: E^m: the entity descriptions at the m-th level, $best\overline{P^M}(m\text{-}1)$: the high-confidence aligned entity pairs at the (m-1)-th level

Output: $H^{\mathrm{s}}(E^m)$: the structural semantic embeddings over entities in E^m; $H^{\mathrm{a}}(E^m)$: the attribute semantic embeddings over each entity in E^m; $H^{\mathrm{c}}(E^m)$: the corroborative embeddings over entities in E^m; $H^{\mathrm{j}}(E^m)$: the joint embeddings over entities in E^m; $H(E^m)$: the comprehensive embeddings over entities in E^m

1. Obtain the equivalent relationships of attribute values according to $best\overline{P^M}(m\text{-}1)$ and complete the attribute triples in KG_1 and KG_2
2. Complete the relational triples in KG_1 and KG_2 according to $best\overline{P^M}(m\text{-}1)$
3. Compute $H^{\mathrm{s}}(E^m)$ according to Eq (3.15)
4. Compute $H^{\mathrm{a}}(E^m)$ according to Eq (3.19)
5. Compute $H^{\mathrm{c}}(E^m)$ according to Eq (3.21)
6. Compute $H^{\mathrm{j}}(E^m)$ according to Eq (3.22)
7. Compute $H(E^m)$ according to Eq (3.23)
8. **return** $\{H^{\mathrm{s}}(E^m), H^{\mathrm{a}}(E^m), H^{\mathrm{c}}(E^m), H^{\mathrm{j}}(E^m), H(E^m)\}$

3.6. Overall algorithm

The pseudo-code of MiAD is illustrated in Algorithm 4. The input of MiAD consists of two domain knowledge graphs with composite entities, denoted KG_1 and KG_2, and the seed set of labeled entity pairs, denoted S_{seed}. The output is a set of the best entity relationship inference modules for entities at different hierarchical levels, the predicted matched entity pairs, the predicted unmatched entity pairs, the new relational triples derived from the intra-level interaction mechanism, and the high-confidence aligned entity pairs. First, MiAD analyzes the hierarchical structure among different categories of entities through the hierarchical structure analyzer (Lines 1–2). Then, MiAD resolves the entity descriptions progressively across hierarchical levels (Lines 3–11). Take the m-th level, for example. For all entities at the m-th level, MiAD generates the multi-perspective semantic features through the cross-level interaction mechanism, constructs the corresponding candidate entity pairs, generates the unified comprehensive embeddings of entities of candidate entity pairs via the spatial mapping, and iteratively optimizes the entity relationship inference module by the mutual reinforcement between the heterogeneous entity relationship inference module and the intra-level interaction mechanism (Lines 5–10).

Algorithm 4 : MiAD

Input: KG_1: the first knowledge graph with composite entities, KG_2: the second knowledge graph with composite entities, S_{seed}: the seed set of labeled entity pairs

Output: $result = \{(bestM^{\text{infer}}(m), bestP^{M}(m), bestP^{UM}(m), bestR^{\text{new}}(m), best\overline{P^{M}}(m))|0 \leq m \leq M\}$

1. get R_1^{HS}, $\{E_1^0, E_1^1, ..., E_1^M\}$ the entities across different levels over KG_1 from the hierarchical structure analyzer
2. get R_2^{HS}, $\{E_2^0, E_2^1, ..., E_2^M\}$ the entities across different levels over KG_2 from the hierarchical structure analyzer
3. Initialize $result \leftarrow \emptyset$
4. **for** $m = 0$ to M **do**
5. Compute $H^{\text{s}}(E_1^m), H^{\text{a}}(E_1^m), H^{\text{c}}(E_1^m), H^{\text{j}}(E_1^m), H(E_1^m), H^{\text{s}}(E_2^m), H^{\text{a}}(E_2^m), H^{\text{c}}(E_2^m), H^{\text{j}}(E_2^m)$, and $H(E_2^m)$ through the cross-level interaction mechanism
6. Obtain candidate entity pairs according to the block filter
7. Compute $H^{\text{u}}(E_1^m)$ and $H^{\text{u}}(E_2^m)$ through the spatial mapping
8. Obtain the best inference model $bestM^{\text{infer}}(m)$, $bestP^{M}(m)$, and $bestP^{UM}(m)$ through the mutual reinforcement between the heterogeneous entity relationship inference module and the intra-level interaction mechanism over the candidate entity pairs at the m-th level
9. $result \leftarrow result \cup (bestM^{\text{infer}}(m), bestP^{M}(m), bestP^{UM}(m), bestR^{\text{new}}(m), best\overline{P^{M}}(m))$
10. **end for**
11. **return** $result$

4. Experiment

In this section, we comprehensively evaluate the effectiveness of our proposed framework, MiAD, through a series of rigorous experiments. (i) We conduct parameter tuning experiments to determine the optimal configuration for hidden layers in the dual-layer structural feature construction and the appropriate candidate set size for the blocker filter. (ii) We conduct comparative experiments with seven state-of-the-art baseline methods to demonstrate the effectiveness of MiAD in addressing complex entity alignment tasks. (iii) We execute ablation studies to analyze the contribution of each component within our framework, particularly emphasizing the significance of the blocker filter. (iv) We evaluate the robustness of MiAD under different dataset scales by conducting sensitivity analysis experiments. In our experimental evaluation, we adopt the standard metrics,namely Hits@n (n = 1, 10, 50) and mean reciprocal rank (MRR), to quantitatively assess the alignment's accuracy, and use recall and the candidate set size ratio (CSSR) to evaluate the performance of the blocker filter. These comprehensive experiments are conducted on a domain-specific dataset (Ta-da) to demonstrate the versatility and effectiveness of our approach in diverse knowledge graph integration scenarios.

4.1. Experimental setup

4.1.1. Experimental dataset

To provide a comprehensive and realistic evaluation of MiAD, we conducted experiments on a pair of Chinese recipe datasets, Ta-da (ZH_MSJ and ZH_ZHYSW). The datasets, characterized by a lack of relational triples, effectively reflect the practical scenarios encountered in domain knowledge graphs and are particularly suitable for validating the effectiveness of our proposed approach. Notably, the dataset has been released on GitHub (https://github.com/DataReconciliation/MiAD) to facilitate access for researchers interested in the complex entity alignment problem. The detailed specifications of the Ta-da dataset are presented in Table 2. #Entities, #Relation-triples, and #Attribute-triples denote the numbers of entities, the relational triples, and attribute triples, respectively.

Table 2. Statistics of the datasets.

Dataset	#Entities	#Relation-triples	#Attribute-triples
ZH_MSJ	8156	1264	95,145
ZH_ZHYSW	6083	896	46,652

4.1.2. Evaluation metrics

We employ common evaluation metrics (i.e., Hits@n and MRR) as the assessment criteria. Hits@n represents the proportion of correct entity alignments ranked within the top n positions among all similarity rankings, as calculated using Eq (4.1). MRR measures the average of reciprocal ranks of correct entity alignments in the similarity rankings, computed according to Eq (4.2). N denotes the total number of test triplets. $Rank_i$ represents the rank position of the correct answer for the i-th triplet; n indicates the top n ranks that we are concerned with; and $I(\cdot)$ is an indicator function, which takes the value of 1 when $Rank_i \leq n$ occurs and 0 otherwise. To comprehensively assess the performance of our proposed method, we utilized the Hits@1, Hits@10, Hits@20, and MRR metrics.

$$Hits@n = \frac{1}{N} \sum_{i=1}^{N} I\left(Rank_i \leq n\right). \tag{4.1}$$

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{Rank_i}. \tag{4.2}$$

In addition to evaluating the model's performance using Hits@n and MRR, assessing the effectiveness of the blocker filter is also essential. To this end, we additionally employ recall and CSSR to provide further insights into the efficiency and accuracy of the candidate set generation process.

Recall and CSSR are employed to evaluate the effectiveness of the blocker filter. CSSR represents the size of candidate set as a proportion of all possible matching entity pairs in KG_1 and KG_2, which can be computed using Eq (4.3). Recall denotes the proportion of correctly matched entity pairs from the set G that are present within the candidate set C, calculated according to Eq (4.4).

$$CSSR = \frac{|C|}{|KG_1 \times KG_2|}. \tag{4.3}$$

$$Recall = \frac{|G \cap C|}{|C|}. \tag{4.4}$$

4.1.3. Baselines

To comprehensively evaluate the effectiveness of MiAD, we perform systematic comparisons across three categories of baseline methods. The first category includes graph neural network-based methods (i.e., GCN-Align [26], MuGNN [27], and HGCN [28]), which effectively capture the structural features and relational semantics of entities. The second category comprises semantic optimization-based methods, including FGWEA [29] and KGAE [30]. FGWEA [29] performs unsupervised entity alignment by optimizing the structural and semantic correspondence between KGs using the Gromov–Wasserstein distance. Unlike FGWEA, KGAE [30] focuses on attribute-level semantics by generating character-level embeddings of attribute values and enhancing entity representations through transitivity-based attribute enrichment. The third category comprises multimodal fusion methods, such as PEEA [31] and SDEA [32], which leverage positional information and pretrained language models to substantially enhance alignment performance. In total, we compare our approach against the following seven representative baseline methods:

• GCN-Align [26], which first employs a GCN to encode both the structural and attribute information of entities, and then utilizes a weighted summation strategy to integrating the joint similarity of the structural and attribute feature vectors, which is a reference criterion for the equivalence relationships across entity descriptions.

• MuGNN [27], which employs GNN to embed the structural information of the knowledge graph into multiple channels. It then uses an attention mechanism to assign weights to the relationships between entities, enhancing the similarity calculations of embedded features to achieve alignment.

• HGCN [28], which employs GCN to capture latent features of entities and relations, iteratively learning their embedded representations through joint learning strategy.

• FGWEA [29], which fully utilizes the structural information of knowledge graphs and optimizes the entities' semantics and the knowledge graphs' structure to obtain a comprehensive comparison of the corresponding entities. This method is an unsupervised entity alignment framework with Gromov–Wasserstein distance.

• KGAE [30], which computes character-level embeddings of the entities' attribute values, and aligns entities by projecting them into a shared semantic space on the basis of their attributes' similarity. KGAE enhances attribute embeddings via a transitivity rule that infers additional attribute triples.

• PEEA [31], which learns the structural and relational information of entities and also integrates positional information during the representation learning process to enhance the connection relationships between entities. This method also has a weakly supervised learning framework.

• SDEA [32], which first utilizes a pretrained language model (transformer) to extract the semantic information from attribute values, then employs gated recurrent unit (GRU) to construct an attention layer that aggregates the structural information from neighboring entities.

4.1.4. Experimental parameter settings

Regarding MiAD, we use 50% of the target data as the test set, 30% of the target data as the training set, and 20% of the target data as the validation set. We implemented the code for seven baselines under identical environments. All the experimental environment's specifications are shown in Table 3. The hyperparameters used in MiAD are listed in Table 4.

Table 3. Experimental environment.

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

Table 4. The model's hyperparameters.

Hyperparameters	Value
Keep-prob	0.9
Learning rate	0.001
Alpha	0.2
Max epoch	100

4.2. Experimental results and analysis

4.2.1. Tunable parameter analysis

MiAD incorporates multiple tunable parameters. The parameter tuning experiment aims to investigate the impact of hidden layer numbers in the dual-layer structure feature construction strategy on the filtering effectiveness and entity alignment results, and the influence of candidate set size on the alignment results. These experiments consider two key factors: The number of hidden layers in the dual-layer structure features and the size of candidate sets generated by the blocker filter. Specifically, we set the number of hidden layers (n) in the dual-layer structure features to range from 1 to 6. MiAD-n in Table 5 represent the MiAD alignment method with n hidden layers in the dual-layer structure features. Additionally, on the basis of the research findings of Xu et al. [33], which demonstrate that the candidate set size significantly impacts the model's performance, we adopted their validated optimal CSSR values (2.53%, 2.81%, and 5.62%) to rapidly achieve the optimal results. This range ensures that the blocker filter achieves the highest recall while generating the smallest possible candidate set. To comprehensively cover all possible experimental configurations, we conducted exhaustive experiments across all combinations, with the results presented in Table 5.

First, we analyzed the dual-layer structure feature construction strategy with varying numbers of hidden layers, examining the recall performance of the blocker filter across different candidate set sizes. The experimental results are shown in Figure 7.

The observations from Figure 7 reveal several key findings. (i) MiAD-1 and MiAD-6 exhibit the lowest performance across various CSSR values. Further investigation reveals that the suboptimal performance of MiAD-1 stems from its limited ability to represent potential features, which is attributed to aggregating an insufficient number of entities. This limitation hinders the effective learning of inter-entity differences. This results in poor filtering effectiveness and lower recall. In contrast, the performance degradation of MiAD-6 is primarily due to excessive entity aggregation, which results in overly similar entity embeddings. This similarity undermines the discriminative capability of the blocker filter, thereby resulting in a noticeable decline in recall performance. (ii) MiAD-2 and MiAD-3 exhibit superior filtering performance across all candidate set sizes. While MiAD-4 and MiAD-5 achieve high recall across different dataset sizes, their performance remains suboptimal compared with MiAD-2 and MiAD-3. (iii) MiAD-3 achieves the highest recall across all CSSR values. These results indicate that blocker filter of MiAD-3 exhibits superior filtering performance.

Table 5. The MRR, $Hits@1$, $Hits@10$, and $Hits@50$ scores of MiAD across different hidden layers and CSSR values.

Hidden layers	CSSR (%)	Recall (%)	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
MiAD-1	2.53	95.12	59.25	92.34	95.56	72.45
	2.81	96.45	59.75	92.67	95.78	73.12
	5.62	97.78	60.50	93.12	96.12	73.78
MiAD-2	2.53	97.77	64.50	94.56	97.23	78.45
	2.81	98.24	65.00	94.78	97.45	78.89
	5.62	99.65	65.50	95.00	97.67	79.34
MiAD-3	2.53	97.80	64.70	94.67	97.34	78.45
	2.81	98.55	65.10	94.89	97.56	79.12
	5.62	99.87	65.50	95.12	97.78	79.56
MiAD-4	2.53	96.87	62.50	93.89	96.45	75.89
	2.81	97.68	63.00	94.12	96.67	76.34
	5.62	98.82	63.50	94.34	96.89	76.78
MiAD-5	2.53	96.98	62.70	94.00	96.67	76.12
	2.81	98.23	63.10	94.23	96.89	76.56
	5.62	98.67	63.50	94.45	97.12	76.89
MiAD-6	2.53	96.01	59.75	92.56	95.67	72.89
	2.81	97.78	60.00	92.78	95.89	73.34
	5.62	97.98	60.50	93.00	96.12	73.78

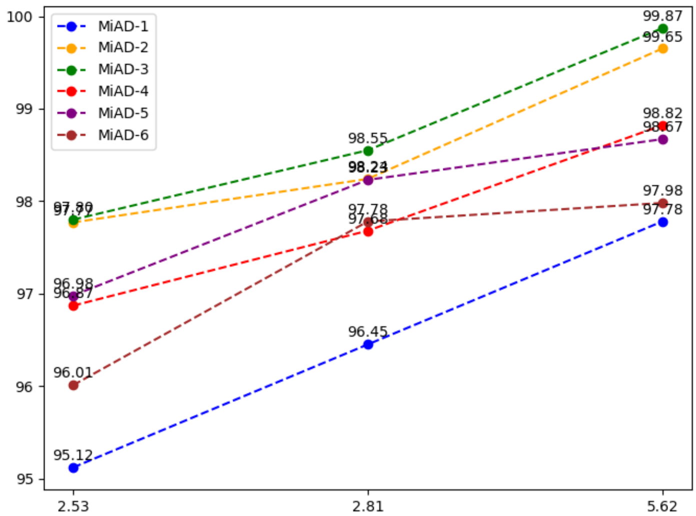

Figure 7. Recall results of the blocker filters with varying numbers of hidden layers and various CSSR values.

Through recall detection using blocker filters on candidate sets of different sizes, it is demonstrated that larger candidate sets enable the model to capture more potential entity matches. Therefore, we conducted experiments on the impact of the number of hidden layers in the dual-layer structure feature construction process on the entity alignment results across different CSSR values. In the experiments, we tested CSSR values of 2.53%, 2.81%, and 5.62%. The experimental results are shown in Figure 8.

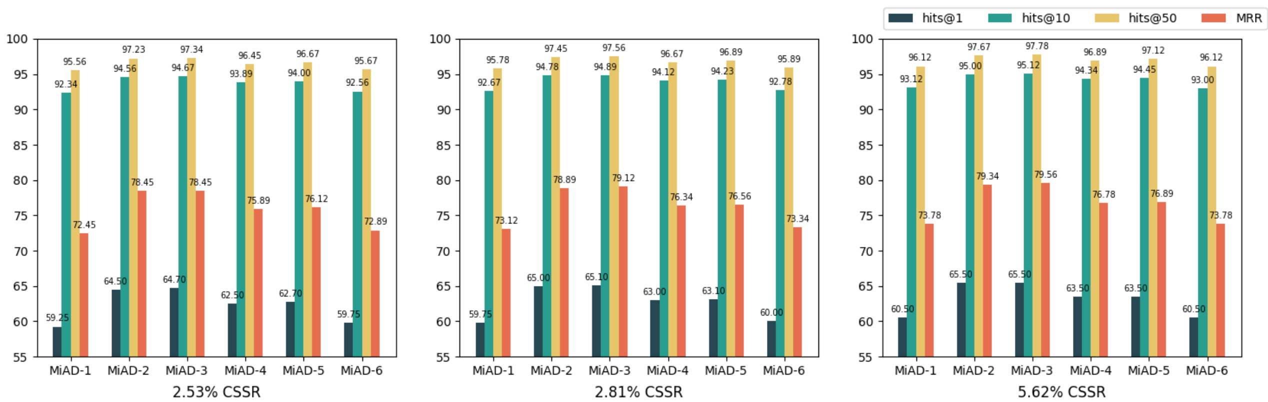

Figure 8. The MRR, $Hits@1$, $Hits@10$, and $Hits@50$ scores of MiAD across different hidden layers and CSSR values.

The experimental results, as shown in Figure 8, reveal two key findings. (i) Among MiAD-1 through MiAD-6, MiAD-6 demonstrates relatively lower performance across all four metrics (Hits@1, Hits@10, Hits@50, and MRR), while MiAD-3 exhibits superior performance. (ii) As the number of hidden layers increases, all performance metrics for MiAD-1 through MiAD-6 show an initial increase followed by a decline, indicating no positive correlation between the number of hidden layers and metric performance. Combining theoretical analysis with experimental results suggests that while increasing the number of hidden layers can enhance the model's fitting capability and feature representation, it may also introduce challenges such as overfitting, gradient-related issues, and increased training time.

On the basis of our comprehensive analysis of the MiAD system's architecture, we observe that the blocker filter must achieve maximum recall with minimal candidate sets, while the alignment phase requires optimal performance across all metrics. The experimental results demonstrate that: (i) In the blocker filter, MiAD-3 consistently achieves superior performance across different CSSR values, reaching a remarkable recall of 99.87% at a CSSR of 5.62%, which highlights its outstanding filtering capability. (ii) In the entity alignment module, MiAD-3 similarly exhibits outstanding alignment performance, achieving the highest scores across almost all metrics for different candidate set sizes. This comprehensive analysis indicates that MiAD-3 represents the optimal configuration for our model. Therefore, we set the number of hidden layers for the dual-layer structural feature construction strategy to 3, with the blocker filter generating a candidate set with a CSSR value of 5.62%, which serves as the baseline model for subsequent experimental comparative analysis.

4.2.2. Performance comparison test

MiAD is a hierarchical division-based entity alignment approach, specifically designed to address the complex entity alignment problem in knowledge graphs with composite entities. The effective implementation of entity hierarchy division in MiAD ensures the accuracy and robustness of entity alignment tasks. We used the Ta-da dataset, which represents a multi-level domain knowledge graph. According to the hierarchical structure analyzer, entity alignment is performed sequentially at each level to optimize the matching of domain knowledge graph pairs. For the complex entity alignment, MiAD utilizes matching information transfer between different entities.

To evaluate the effectiveness of MiAD, we compared it with five representative entity alignment methods: GCN-Align, FGWEA, PEEA, SDEA, and KGAE. Among them, GCN-Align and SDEA are structure-based approaches that utilize graph convolutional networks, while FGWEA and PEEA focus on embedding generation using different similarity aggregation mechanisms. KGAE, in particular, emphasizes attribute information and generates embeddings through attribute character-level encoding. We did not adopt the strategy of replacing relational triples with attribute triples for embedding learning. Instead, the original relational triples are retained for all models to maintain consistency and ensure fairness in the utilization of structural information. Due to the structural sparsity of the Ta-da dataset, structurally dependent models such as MuGNN and HGCN are excluded from the comparison. All experiments are conducted under identical conditions to address the same alignment task on the Ta-da dataset. The comparative experimental results are presented in Table 6.

Table 6. Experimental comparison with baseline methods.

Method	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
GCN-Align	46.25	86.74	92.21	60.07
FGWEA	55.94	88.79	92.36	69.45
PEEA	53.59	88.61	90.16	67.98
SDEA	55.18	89.26	91.65	70.94
KGAE	60.10	92.17	93.52	74.54
MiAD	**65.74**	**98.35**	**99.02**	**80.33**

The results in Table 6 demonstrate that MiAD outperforms all baseline methods across all evaluation metrics, achieving a Hits@1 of 65.74%, a Hits@10 of 98.35%, and an MRR of 80.33%. Compared with KGAE, the best-performing baseline, MiAD achieves improvements of 5.64% in Hits@1, 6.18% in Hits@10, and 5.79% in MRR, demonstrating its superiority in handling structurally sparse data.

Analysis of these results reveals four key findings. (i) The attribute semantic plays a significant role in addressing structurally sparse data. Compared with the other baseline methods, attribute-aware methods, i.e., KGAE and MiAD, achieve superior matching performance across all evaluation metrics. (ii) The most significant distinction between MiAD and the five baseline methods lies in its filter implementation. By effectively eliminating interfering and mismatched entities, MiAD exhibits improved alignment capability. Notably, its filtering module leverages a dual-layer feature construction strategy, enabling robust filtering based on entity attributes even in the absence of structural information—an advantage not offered by the baseline methods. (iii) Regarding the embedding module, MiAD incorporates four feature dimensions—structural, attributive, corroborative and joint information—for embedding learning, constructing multi-perspective entity features to comprehensively capture the latent entity characteristics. This enables MiAD to effectively complete alignment tasks even when certain features are missing. (iv) MiAD employs an iterative optimization strategy, progressively refining the alignment results for more effective task completion.

Through the in-depth analysis of the design principles of baseline methods, it can be determined that all five methods accomplish entity alignment tasks based on complete knowledge graphs, and regard entity structural information as an important component of the model. However, their effectiveness is constrained when confronting the prevalent entity relationship sparsity in domain knowledge graphs. Moreover, domain knowledge graphs contain numerous high-confidence entity matching interactions during alignment, making the effective elimination of such interference a critical challenge in achieving optimal results. Five baseline methods require multiple initial conditions during execution, where insufficient or missing initial conditions may impact the alignment outcomes to varying degrees, reflecting challenges in their applicability and robustness.

4.2.3. Ablation study

To systematically evaluate the contribution of each key component in the MiAD framework, we conduct a comprehensive ablation study by removing five core modules: the blocker filter, the structural semantic aggregator, the attribute semantic aggregator, the corroborative aggregator, and the joint aggregator. The resulting model variants are denoted as MiAD-f, MiAD-s, MiAD-a, MiAD-c, and MiAD-j, respectively. The performance of each ablated model is measured using four valuation metrics: Hits@1, Hits@10, Hits@50, and MRR. The experimental results are presented in Table 7.

Table 7. Experimental results of the ablation analysis.

Method	Hits@1 (%)	Hits@10 (%)	Hits@50 (%)	MRR (%)
MiAD-f	55.42	94.35	97.98	70.44
MiAD-s	56.23	94.73	97.78	72.12
MiAD-a	58.94	96.64	98.09	76.88
MiAD-c	59.13	97.92	98.23	79.73
MiAD-j	55.56	93.89	97.50	70.68
MiAD	**65.74**	**98.35**	**99.02**	**80.33**

1) Effect of the blocker filter. Comparative experiments between MiAD and MiAD-f were conducted to validate the necessity of the blocker filter. Table 7 reveals that MiAD outperforms MiAD-f across all metrics. Specifically, compared with MiAD-f, MiAD achieves performance improvements of 10.32%, 4.0%, 1.04%, and 9.89% in Hits@1, Hits@10, Hits@50, and MRR, respectively. These results indicate that MiAD effectively eliminates noisy entities and mismatched entities, and significantly enhances the alignment performance.

2) Effect of various semantic aggregators. As reported in Table 7, the ablation study reveals the individual impact of each semantic aggregator by evaluating four MiAD variants, each with one component removed. The analysis leads to the following key observations: (i) The joint aggregator plays the most critical role in achieving effective entity alignment. Removing it (MiAD-j) leads to a 10.18% drop in Hits@1 and a 9.65% decrease in MRR, indicating that the joint aggregator is indispensable for integrating multi-perspective semantic representations into a unified, same-dimensional space. (ii) The structural aggregator makes a substantial contribution to the alignment performance. When the structural aggregator is excluded, MiAD-s suffers a 9.51% reduction in Hits@1 and an 8.21% decline in MRR, demonstrating the importance of structural cues derived from relational and multi-hop connections. (iii) The attribute and corroborative aggregators serve as auxiliary components. MiAD-a and MiAD-c result in comparatively smaller performance drops (for example, MiAD-c decreases MRR by only 0.6%) which suggests that while these aggregators enhance the semantic richness, they are less dominant than their structural and joint counterparts. Overall, the results validate that all four aggregators play complementary roles in improving the alignment performance.

4.2.4. Sensitivity analysis

The sensitivity analysis experiments aim to evaluate the robustness and scalability of MiAD in datasets of different scales. This experiment is crucial for assessing whether the model maintains consistent performance regardless of the knowledge graph scale, which is particularly important for real-world applications where domain knowledge graphs can vary significantly in size. By evaluating the model on three progressively larger datasets (KG-1K, KG-3K, and KG-5K), we can systematically investigate the effect of dataset size on the entity alignment performance of MiAD.

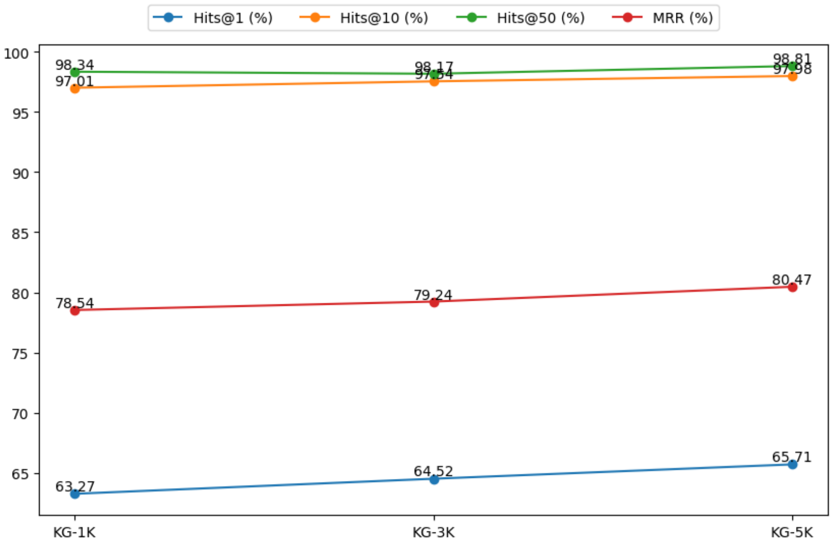

Figure 9. Comparison of recall rates.

According to the previous experiments, MiAD-3 demonstrated superior alignment performance across all tests, and was therefore selected as the test model for analyzing sensitivity to dataset size. We conducted experiments using three datasets — KG-1K, KG-3K, and KG-5K — containing 1K, 3K, and 5K entity pairs, respectively, to evaluate whether the model maintains robust entity alignment performance across different dataset sizes. The experimental results are presented in Figure 9.

The experimental results depicted in Figure 9 demonstrate that MiAD-3 exhibits remarkable stability across datasets of different sizes. With an increase in the scale of the dataset, performance metrics such as Hits@10, Hits@50, and MRR demonstrate consistent trends, indicating that the model has a strong generalization ability. Although Hits@1 displays slight fluctuations compared with other metrics, it maintains an overall stable trajectory and shows a gradual improvement with larger datasets. This upward trend across all metrics as the dataset size increases confirms that MiAD effectively leverages additional training examples to enhance the precision of its entity alignment.

The observed performance improvement with larger datasets can be attributed to two primary factors. First, more extensive datasets provide richer semantic contexts and more diverse entity relationships, enabling the model to learn more comprehensive feature representations. Second, larger knowledge graphs offer more examples of complex hierarchical relationships between entities at different atomic levels, allowing MiAD to better capture the matching information interaction patterns that are fundamental to its design. This scalability characteristic is particularly valuable for practical applications in domain-specific knowledge graph integration, where the ability to maintain high alignment accuracy across varying data volumes is essential.

5. Related work

Entity alignment, as a key technology in data integration, has attracted widespread attention from scholars in recent years. Most of the state-of-art entity alignment methods utilize representation learning, aiming to transform input information into low-dimensional vectors while fully preserving its rich features. In knowledge graphs, representation learning can be used to map the relationships between entities and descriptive information into vector spaces, effectively capturing hidden semantic relationships and structural features among entities.

Existing state-of-the-art solutions with representation learning for completing entity alignment tasks are mainly divided into two categories: translation-based methods and graph-embedding-based methods.

5.1. *Translation-based methods*

Translation-based methods model relational triples as translational operations in continuous vector spaces and have been extensively studied for cross-lingual entity alignment. These methods are typically divided into two categories: (i) foundational methods that aims to effectively learn the entity and relation embeddings [34–38], and (ii) enhanced methods that refine these embeddings by auxiliary mechanisms such as the interactions between entities and relationships, temporal dynamics, or hybrid projection mechanisms [39–44].

Several foundational methods have been proposed, including the following representative approaches. Initially, researchers proposed the structured embedding (SE) model [34], which constructs distinct vector spaces for different relationships and accomplishes entity alignment by computing the distance L_1 between the projected vectors of entities within these relationship-specific spaces. TransE [35] is a classic graph representation learning method that represents entity relationships using relational triples (h, r, t). TransE maps the semantic relationships r between entities as translational relationships between the head entity h and the tail entity t embedding vectors in space, determining entities' similarity by calculating whether their relationships satisfy the constraint $t \approx h + r$. Although TransE can effectively capture entities' relationships through translational invariance [36] in word vector spaces, it assumes that all relationships are linear translations in vector space. This limitation restricts TransE to effectively capturing only simple one-to-one relationships, making it less effective for alignment tasks involving one-to-many or many-to-many complex relationships. To overcome TransE's limitations in handling complex relationships, TransH [37] introduces a point-wise space, assigning distinct projection planes for different relationship types. It determines entities' alignment relationships by calculating whether the relation r and head–tail entity projections $\mathrm{h}_\perp$ and $\mathrm{t}_\perp$ on hyperplane r satisfy TransE's constraint conditions $t_\perp \approx \mathrm{h}_\perp + r$. Both TransE and TransH map entities and relationships to the same space, ignoring the inherent distinctions between entities and relationships in knowledge graphs. To address this issue, TransR [38] embeds entities and relationships in different vector spaces, then uses a projection matrix M_r to map entity embeddings to the relationship space r, evaluating entity matching through the scoring function $f_r(h, t) = -\|h_\perp + r - t_\perp\|_2^2$.

Although TransR uses projection matrices to embed entities and relationships in different vector spaces, these matrices are generated solely on the basis of relationships, overlooking the interactions between entities and relationships. Addressing this issue, TransD [39] dynamically generates the corresponding projection matrices M_{rh} and M_{rt} using head entities, tail entities, and relationships. Moreover, its strategy of constructing separate projection matrices for head and tail entities effectively distinguishes how different entity types and attributes influence the embedding representations. While these methods focus on transforming entity or relationship mappings, they do not account for the varying impact of that different relationships may have on entity features. TransM [40] tackles this issue by introducing a relationship weight w_r, assigning lower weights to complex entity relationships (1-N, N-1, etc.), effectively handling entity alignment tasks with complex relationships. These methods are single approaches to entity alignment tasks. In contrast, STransE [41], a composite method, combines SE with TransE, solving two projection matrices for each relationship to map head and tail entities to the relationship space, completing entity alignment by calculating the distance between these entities within the relationship space. During alignment, in addition to focusing on internal entity information, some approaches, like HyTE [42], TA-TransE [43], and TTransE [44], incorporate temporal information as vectors into the entity alignment, improving performance in some cases.

Although translation-based methods including foundational methods and enhanced methods, achieve good results in cross-lingual entity alignment tasks, they rely on the availability of complete knowledge graphs. Domain knowledge graphs often lack comprehensive structural and attribute information, making translation-based models unsuitable for domain knowledge graph entity alignment tasks. Furthermore, translation-based models primarily rely on local semantic information from relational triples, often failing to effectively capture the global semantic information present in knowledge graphs.

5.2. Graph-embedding-based methods

Graph-embedding-based methods map the entities and relationships of knowledge graphs into low-dimensional vector spaces, learning the complex relationships and semantic features between entities through graph neural networks for entity alignment. Graph-embedding-based methods optimize knowledge graph embeddings from single or multiple perspectives, including graph structure, attributes, the entities' names, or entity descriptions. These methods are typically grouped into two categories: (i) single-perspective models that focus on structure or attributes [27, 33, 45–49], (ii) dual-perspective models that integrate both structure and attributes information [24, 26, 50–52], (iii) multi-perspective models designed to incorporate structural features, the entities' names, entity descriptions, or other information [28, 53–57].

Representative single-perspective models typically focus on either structural information or attribute information to learn entity embeddings. For the structural information of knowledge graphs, MuGNN [27] utilizes GNN to model structural information and employs two weighting mechanisms – self-attention and cross-KG attention – to assign weights to entity relationships, thus forming multiple channels. Entity alignment is achieved by calculating the similarity of multi-channel joint entity features. VR-GCN [45] improves the entity feature update mechanism of the original GCN by integrating the translational invariance principle proposed in TransE. In VR-GCN, the entity features are updated by aggregating embedding representations of adjacent entities and relationships, while the relationship embeddings are updated through head and tail entity embeddings, effectively capturing the structural information of the knowledge graph. Similar approaches that leverage structural information to construct entity features include SelfAttention-GCN [46], MRAEA [47], and RREA [48]. In addition to models based on the structural information of knowledge graphs, some methods focus on entity attribute information for alignment tasks. Schema-Agnostic [49] first employs a filtering mechanism to select candidate entity pairs, then connects entities' attributes and utilizes a pretrained BERT classifier to predict labels (match or non-match). Similarly, DomainEA [33] first constructs a blocking filtering mechanism based on entities' attributes to generate candidate sets for each entity. To address the lack of structural information between domain knowledge graph entities, DomainEA uses attribute triples instead of relational triples with candidate sets. It learns knowledge graphs' attribute features and indirect structural features to complete entity alignment by calculating the similarity of joint embedding features. These methods mostly learn the entities' feature embeddings from single perspectives like structure or attributes.

To address the inherent limitations of single-perspective models, researchers have proposed dual-perspective entity alignment strategies that integrate complementary information from both structural and attribute aspects, acknowledging that each provides unique and indispensable signals. GCN-Align [26] is the first to introduce GCNs into knowledge graph entity alignment tasks, learning the global features and weight matrices from both the structural and attribute perspectives, completing alignment by calculating the weighted feature similarity between entities. AttrGNN [24] first employs graph partitioning to segment knowledge graphs into four distinct types of subgraphs on the basis of their entity attribute categories. Then it uses an attributed value encoder and mean aggregator composed of a GNN to learn the entities' attributes and structural view features on each subgraph. Finally, AttrGNN integrates embedding vectors from both perspectives to generate the final entity embedding features, completing alignment tasks by computing the entity embedding features' similarity within a unified vector space. Other representative approaches, such as SEEA [50], LinkNBed [51], and CG-MuAlign [52], also follow the multi-perspective by jointly modeling structural and attribute features to construct more comprehensive entity embeddings for alignment tasks.

Structural and attribute information, as intrinsic entity features, can effectively mine hidden entity features if reasonably used with complete knowledge graphs. However, in many application scenarios, obtaining complete knowledge graphs is challenging, and relying solely on internal entity information makes it difficult to achieve high-precision entity alignment results. Therefore, researchers have introduced external entity information to assist entity alignment, such as HGCN [28] and RDGCN [53] incorporating information on the entities' names, and HMAN [54] incorporating entity description information. DSGNet [55] proposes a decoupled semantic GNN that projects entities into multiple independently modeled semantic subspaces to capture semantic information. A relation-aware top-k sampling strategy assigns weights on the basis of neighbor relevance to filter out noisy neighbors, enabling graph-level semantic decoupling and leading to higher-quality embeddings. GSIEA [56] proposes a graph structure prefix injection transformer for multi-modal entity alignment. It integrates structural information with additional modalities such as attributes and images by aggregating the structural features of neighboring entities and injecting them as prefix representations into the transformer encoder, thereby explicitly modeling structural heterogeneity across knowledge graphs. AdaGCN [57] introduces an adaptive message passing mechanism that employs semantic-aware gating to dynamically aggregate or separate neighbor embeddings. By regulating the information flow according to relevance, it mitigates oversmoothing and improves the discriminative capability of entity representations. This fine-grained control provides a more flexible alternative to conventional fixed-layer GCNs.

In general, knowledge graph entity alignment is a complex task that requires the consideration of multiple factors, including the structure and attribute information of knowledge graphs, the incorporation of external information, and the modeling of complex relationships. The current research challenge lies in exploring an entity alignment approach that can better utilize this information to improve both the accuracy and efficiency of entity alignment.

6. Conclusions

In summary, we propose a MiAD entity alignment approach for addressing entity alignment challenges in domain-specific knowledge graphs. MiAD leverages hierarchical structure analysis and iterative optimization through intra-level and cross-level interactions to enhance entity relationship inference. A multi-channel embedding method with four aggregators captures comprehensive entity semantics from multiple perspectives. Extensive experiments demonstrate that the MiAD has superior performance, achieving a 99.87% recall. Future work will focus on extending the approach to more diverse entity types and integrating additional semantic features for real-world applications.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62102372), the School-Level Young Backbone Teacher Training Program of Zhengzhou University of Light Industry, the project of Science and Technology in Henan Province (Nos. 232102210078 and 242102210037), and the Doctor Scientific Research Fund of Zhengzhou University of Light Industry (No. 2021BSJJ029). Yaoli Xu, Tong Han, and Haojie Zhai these authors contributed equally to this work.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. T. C. Zhang, X. Tian, X. H. Sun, M. H. Yu, Y. H. Sun, G. Yu, Overview on knowledge graph embedding technology research (in Chinese), *J. Software*, **34** (2023), 277–311. https://doi.org/10.13328/j.cnki.jos.006429
2. L. A. Huang, X. Luo, EASA: Entity alignment algorithm based on semantic aggregation and attribute attention, *IEEE Access*, **8** (2020), 18162–18170. https://doi.org/10.1109/access.2020.2968620
3. J. Chicaiza, P. Valdiviezo-Diaz, A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions, *Information*, **12** (2021), 232. https://doi.org/10.3390/info12060232
4. L. Y. Yu, Z. Guo, G. Chen, X. X. Zhang, Y. W. Tang, H. Wei, Question answering system based on knowledge graph in air defense field, *J. Phys. Conf. Ser.*, **1693** (2020), 012033. https://doi.org/10.1088/1742-6596/1693/1/012033

5. M. Wang, H. F. Wang, B. H. Li, X. Zhao, X. Wang, Survey on key technologies of new generation knowledge graph (in Chinese), *J. Comput. Res. Dev.*, **59** (2022), 1947–1965. https://doi.org/10.7544/issn1000-1239.20210829

6. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in *The Semantic Web*, Springer, (2007), 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

7. F. M. Suchanek, G. Kasneci, G. Weikum, YAGO: A large ontology from Wikipedia and WordNet, *J. Web Semant.*, **6** (2008), 203–217. https://doi.org/10.1016/j.websem.2008.06.001

8. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A collaboratively created graph database for structuring human knowledge, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, (2008), 1247–1250. https://doi.org/10.1145/1376616.1376746

9. D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, *Commun. ACM*, **57** (2014), 78–85. https://doi.org/10.1145/2629489

10. R. Navigli, S. P. Ponzetto, BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, *Artif. Intell.*, **193** (2012), 217–250. https://doi.org/10.1016/j.artint.2012.07.001

11. Y. J. Gao, C. C. Ge, Y. X. Guo, L. Chen, Survey on data integration technologies for relational data and knowledge graph (in Chinese), *J. Software*, **34** (2023), 2365–2391. https://doi.org/10.13328/j.cnki.jos.006808

12. H. R. Huang, C. Li, X. T. Peng, L. F. He, S. Guo, H. Peng, et al., Cross-knowledge-graph entity alignment via relation prediction, *Knowledge-Based Syst.*, **240** (2022), 107813. https://doi.org/10.1016/j.knosys.2021.107813

13. Y. L. Xu, Z. H. Li, Q. Chen, Y. Y. Wang, F. F. Fan, An approach for reconciling inconsistent pairs based on factor graph (in Chinese), *J. Comput. Res. Dev.*, **57** (2020), 175–187. https://doi.org/10.7544/issn1000-1239.2020.20180691

14. H. C. Wang, Y. N. Wang, J. F. Li, T. Luo, Degree aware based adversarial graph convolutional networks for entity alignment in heterogeneous knowledge graph, *Neurocomputing*, **487** (2022), 99–109. https://doi.org/10.1016/j.neucom.2022.02.002

15. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907. https://doi.org/10.48550/arXiv.1609.02907

16. L. Yao, C. S. Mao, Y. Luo, KG-BERT: BERT for knowledge graph completion, preprint, arXiv:1909.03193. https://doi.org/10.48550/arXiv.1909.03193

17. L. B. Soares, N. Fitzgerald, J. Ling, T. Kwiatkowski, Matching the blanks: Distributional similarity for relation learning, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (2019), 2895–2905. https://doi.org/10.18653/v1/P19-1279

18. P. L. H. Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in *Findings of the Association for Computational Linguistics: EMNLP 2021*, (2021), 2370–2381. https://doi.org/10.18653/v1/2021.findings-emnlp.204

19. K. W. Church, Word2Vec, *Nat. Lang. Eng.*, **23** (2017), 155–162. https://doi.org/10.1017/S1351324916000334

20. P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, *Trans. Assoc. Comput. Ling.*, **5** (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

21. S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in *5th International Conference on Learning Representations*, 2019.

22. M. H. Chen, Y. T. Tian, K. W. Chang, S. Skiena, C. Zaniolo, Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment, in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, (2018), 3998–4004. https://doi.org/10.24963/ijcai.2018/556

23. J. Liu, B. Chai, Z. Shang, A cross-lingual medical knowledge graph entity alignment algorithm based on neural tensor network, *Basic Clin. Physiol. Pharmacol. Toxicol.*, **128** (2021), 31–32.

24. Z. Y. Liu, Y. X. Cao, L. M. Pan, J. Z. Li, Z. Y. Liu, T. S. Chua, Exploring and evaluating attributes, values, and structures for entity alignment, preprint, arXiv:2010.03249. https://doi.org/10.48550/arXiv.2010.03249

25. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, (2017), 6000–6010.

26. Z. C. Wang, Q. S. Lv, X. H. Lan, Y. Zhang, Cross-lingual knowledge graph alignment via graph convolutional networks, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (2018), 349–357. https://doi.org/10.18653/v1/d18-1032

27. Y. Cao, Z. Y. Liu, C. J. Li, J. Z. Li, T. S. Chua, Multi-channel graph neural network for entity alignment, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (2019), 1452–1461. https://doi.org/10.18653/v1/p19-1140

28. Y. T. Wu, X. Liu, Y. S. Feng, Z. Wang, D. Y. Zhao, Jointly learning entity and relation representations for entity alignment, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, (2019), 240–249. https://doi.org/10.18653/v1/d19-1023

29. J. H. Tang, K. F. Zhao, J. Li, A fused gromov-wasserstein framework for unsupervised knowledge graph entity alignment, in *Findings of the Association for Computational Linguistics: ACL 2023*, (2023), 3320–3334. https://doi.org/10.18653/v1/2023.findings-acl.205

30. B. D. Trisedya, J. Qi, R. Zhang, Entity alignment between knowledge graphs using attribute embeddings, in *Proceedings of the 33th AAAI Conference on Artificial Intelligence and the 31th Innovative Applications of Artificial Intelligence Conference and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, (2019), 297–304. https://doi.org/10.1609/aaai.v33i01.3301297

31. W. Tang, F. L. Su, H. F. Sun, Q. Qi, J. Y. Wang, S. M. Tao, et al., Weakly supervised entity alignment with positional inspiration, in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, (2023), 814–822. https://doi.org/10.1145/3539597.3570394

32. Z. Y. Zhong, M. H. Zhang, J. Fan, C. X. Dou, Semantics driven embedding learning for effective entity alignment, in *2022 IEEE 38th International Conference on Data Engineering*, (2022), 2127–2140. https://doi.org/10.1109/icde53745.2022.00205

33. Y. L. Xu, J. J. Zhong, S. Z. Zhang, C. L. Li, P. Li, Y. B. Guo, et al., A domain-oriented entity alignment approach based on filtering multi-type graph neural networks, *Appl. Sci.*, **13** (2023), 9237. https://doi.org/10.3390/app13169237

34. A. Bordes, J. Weston, R. Collobert, Y. Bengio, Learning structured embeddings of knowledge bases, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **25** (2011), 301–306. https://doi.org/10.1609/aaai.v25i1.7917

35. A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, **2** (2013), 2787–2795.

36. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, (2013), 3111–3119.

37. Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **28** (2014), 1112–1119. https://doi.org/10.1609/aaai.v28i1.8870

38. Y. Lin, Z. Y. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **29** (2015), 2181–2187. https://doi.org/10.1609/aaai.v29i1.9491

39. G. Ji, S. Z. He, L. H. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, **1** (2015), 687–696. https://doi.org/10.3115/v1/p15-1067

40. M. Fan, Q. Zhou, E. Chang, T. F. Zheng, Transition-based knowledge graph embedding with relational mapping properties, in *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, Department of Linguistics, Chulalongkorn University, (2014), 328–337.

41. D. Q. Nguyen, K. Sirts, L. Qu, M. Johnson, StransE: A novel embedding model of entities and relationships in knowledge bases, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (2016), 460–466. https://doi.org/10.18653/v1/N16-1054

42. S. S. Dasgupta, S. N. Ray, P. Talukdar, HyteTE: Hyperplane-based temporally aware knowledge graph embedding, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (2018), 2001–2011. https://doi.org/10.18653/v1/d18-1225

43. A. García-Durán, S. Dumančić, M. Niepert, Learning sequence encoders for temporal knowledge graph completion, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (2018), 4816–4821. https://doi.org/10.18653/v1/d18-1516

44. J. Leblay, M. W. Chekol, Deriving validity time in knowledge graph, in *Companion Proceedings of the The Web Conference*, (2018), 1771–1776. https://doi.org/10.1145/3184558.3191639

45. R. Ye, X. Li, Y. J. Fang, H. Y. Zang, M. Z. Wang, A vectorized relational graph convolutional network for multi-relational network alignment, in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, (2019), 4135–4141. https://doi.org/10.24963/ijcai.2019/574

46. J. Chen, Z. X. Li, P. P. Zhao, A. Liu, L. Zhao, Z. G. Chen, et al., Learning short-term differences and long-term dependencies for entity alignment, in *The Semantic Web – ISWC 2020*, Springer, **12506** (2020), 92–109. https://doi.org/10.1007/978-3-030-62419-4_6

47. X. Mao, W. T. Wang, H. M. Xu, M. Lan, Y. B. Wu, MRAEA: An efficient and robust entity alignment approach for cross-lingual knowledge graph, in *Proceedings of the 13th International Conference on Web Search and Data Mining*, (2020), 420–428. https://doi.org/10.1145/3336191.3371804

48. X. Mao, W. T. Wang, H. M. Xu, Y. B. Wu, M. Lan, Relational reflection entity alignment, in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, (2020), 1095–1104. https://doi.org/10.1145/3340531.3412001

49. K. S. Teong, L. K. Soon, T. T. Su, Schema-agnostic entity matching using pre-trained language models, in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, (2020), 2241–2244. https://doi.org/10.1145/3340531.3412131

50. S. P. Guan, X. L. Jin, Y. Z. Wang, Y. T. Jia, H. W. Shen, Z. X. Li, et al., Self-learning and embedding based entity alignment, *Knowl. Inf. Syst.*, **59** (2019), 361–386. https://doi.org/10.1007/s10115-018-1191-0

51. R. Trivedi, B. Sisman, X. L. Dong, F. Christos, J. Ma, H. Y. Zha, LinkNBed: Multi-graph representation learning with entity linkage, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, (2018), 252–262. https://doi.org/10.18653/v1/p18-1024

52. Q. Zhu, H. Wei, B. Sisman, D. Zheng, C. Faloutsos, X. L. Dong, et al., Collective multi-type entity alignment between knowledge graphs, in *Proceedings of The Web Conference*, (2020), 2241–2252. https://doi.org/10.1145/3366423.3380289

53. Y. T. Wu, X. Liu, Y. S. Feng, Z. Wang, R. Yan, D. Y. Zhao, Relation-aware entity alignment for heterogeneous knowledge graphs, in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, (2019), 5278–5284. https://doi.org/10.24963/ijcai.2019/733

54. H. W. Yang, Y. Y. Zou, P. Shi, W. Lu, J. Lin, X. Sun, Aligning cross-lingual entities with multi-aspect information, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, (2019), 4431–4441. https://doi.org/10.18653/v1/d19-1451

55. Z. F. Li, W. Huang, X. C. Gong, X. Y. Luo, K. Xiao, H. L. Deng, et al., Decoupled semantic graph neural network for knowledge graph embedding, *Neurocomputing*, **611** (2025), 128614. https://doi.org/10.1016/j.neucom.2024.128614

56. Y. Zhang, X. Luo, J. Hu, M. Zhang, K. Xiao, Z. Li, Graph structure prefix injection transformer for multi-modal entity alignment, *Inf. Process. Manage.*, **62** (2025), 104048. https://doi.org/10.1016/j.ipm.2024.104048

57. Z. F. Li, L. F. Chen, Y. Jian, H. Wang, Y. Zhao, M. Zhang, et al., Aggregation or separation? Adaptive embedding message passing for knowledge graph completion, *Inf. Sci.*, **691** (2025), 121639. https://doi.org/10.1016/j.ins.2024.121639

AIMS Press