LLM-Align: Utilizing Large Language Models for Entity Alignment in Knowledge Graphs

Xuan Chen[1], Tong Lu[1], Zhichun Wang[1,2†]

[1]School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China
[2]Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education, Beijing 100875, China

Keywords: Entity alignment; Large language model; Attribute selection; Relation Selection; Multi-round Voting Mechanism

Citation: Chen X., Lu T., Wang Z.C.: LLM-Align: Utilizing Large Language Models for Entity Alignment in Knowledge Graphs. Data Intelligence, Vol. XX, Art. No.: 2025??XX, pp. 1–24, 2025. DOI: https://doi.org/10.3724/2096-7004.di.2025.0072

ABSTRACT

Entity Alignment (EA) seeks to identify and match corresponding entities across different Knowledge Graphs (KGs), playing a crucial role in knowledge fusion and integration. Embedding-based entity alignment (EA) has recently gained considerable attention, resulting in the emergence of many innovative approaches. Initially, these approaches concentrated on learning entity embeddings based on the structural features of knowledge graphs (KGs) as defined by relation triples. Subsequent methods have integrated entities' names and attributes as supplementary information to improve the embeddings used for EA. However, existing methods lack a deep semantic understanding of entity attributes and relations. In this paper, we propose a Large Language Model (LLM) based Entity Alignment method, LLM-Align, which explores the instruction-following and zero-shot capabilities of Large Language Models to infer alignments of entities. LLM-Align uses heuristic methods to select important attributes and relations of entities, and then feeds the selected triples of entities to an LLM to infer the alignment results. To guarantee the quality of alignment results, we design a multi-round voting mechanism to mitigate the hallucination and positional bias issues that occur with LLMs. Experiments on three EA datasets, demonstrating that our approach achieves state-of-the-art performance compared to existing EA methods.

1. INTRODUCTION

Knowledge Graphs (KGs) represent structured information of real-world entities, which are widely employed in research fields such as information retrieval [1-2], recommendation systems [3-4], image

† Corresponding author: Zhichun Wang (E-mail: zcwang@bnu.edu.cn, ORCID: 0000-0001-8797-7065).

classification [5-6], etc. Most knowledge graphs are developed independently by various organizations, using diverse data sources and languages. As a result, KGs often exhibit heterogeneity, where the same entity may appear in different KGs with varying representations. However, KGs can also complement each other, as information about a single entity may be spread across multiple graphs. To address this heterogeneity and integrate knowledge from different KGs, it's crucial to perform Entity Alignment (EA), which involves matching entities across separate KGs.

The problem of EA has been studied for years, many EA approaches have been proposed. Recently, embedding-based EA has gained considerable attention. Embedding-based EA approaches first learn low-dimensional vector representations of entities, and then match entities in vector spaces. According to the used embedding techniques, embedding-based EA approaches mainly fall into two groups: Translation-based approaches and Graph Neural Network (GNN)-based approaches. Translation-based approaches learn entity embeddings using TransE [7] and its extensions, such as MTransE [8], JAPE [9], and BootEA [10]. GNN-based approaches generate neighborhood-aware entity representations by aggregating the features of their neighbors, representative approaches include GCN-Align [11], MuGNN [12], and AliNet [13], et al. To further improve the EA results, some approaches explored entities' attribute information to enhance the entity embeddings, including MultiKE [14], AttrGNN [15] and CEA [16], etc. However, most existing approaches use different techniques to encode information from relational and attributive triples.

Recently, Large Language Models (LLMs) have demonstrated outstanding capabilities in factual question answering [17], arithmetic reasoning [18], logical reasoning [19]. LLMs hold significant potential for enhancing the understanding of entity relationships and attribute information, facilitating more accurate entity alignment. Several approaches have already proposed to utilize LLMs in EA tasks. AutoAlign [20] employs LLMs to integrate the obtained entity type information as a supervision signal into traditional structure-based methods for joint training. LLMEA [21] integrates knowledge from both KGs and LLMs to align entities. ChatEA [22] explores the LLMs' capability for multi-step reasoning to enhance the accuracy of EA. While LLMs have demonstrated their capabilities in EA tasks, there remain challenging issues when applying LLMs to EA:

- KGs usually contain a large number of both attributive and relational triples of entities, not all of these triples are useful and important for aligning entities. If we take all the triples of related entities as inputs to LLMs, irrelevant triples might disturb the reasoning process of LLMs.
- EA tasks can be formatted as judgment questions or multi-choice selection questions for LLMs. The single round decisions made by LLMs sometimes are not reliable, leading to unsatisfying EA results.

To solve the above challenges, we proposed an LLM-based Entity Alignment framework, LLM-Align. LLM-Align first uses any existing EA model as candidate selector, which computes similarities between entities in source and target KGs, and gets candidate alignments by selecting top-k nearest neighbors from the target KG for source entities. Then LLM-Align employs LLMs' reasoning abilities to get the alignment results from the candidates. Specifically, contributions of this work include:

- We propose a three-stage EA framework using LLMs. First, alignment candidates are selected using any existing EA model. Next, attribute-based reasoning and relation-based reasoning are performed sequentially, with LLMs utilizing both relational and attributive triples of entities in a consistent manner.
- We propose heuristic attribute and relation selection methods for LLM-based EA, which select the most informative attributes and relationships of entities to create concise, effective prompts for LLMs.
- We design a multi-round voting mechanism for LLMs to generate reliable EA results. By reordering the input candidates and let LLMs vote for the target entities multiple times, more accurate EA results can be obtained by overcoming the hallucination and positional bias problems in LLMs.
- We conduct comprehensive experiments on real-world EA datasets, comparing LLM-Align with several recent approaches. Results demonstrate that LLM-Align effectively enhances the EA performance of both strong and weaker models. When combined with strong models, LLM-Align achieves the best results among all approaches compared.

The rest of this paper is organized as follows: Section 2 discusses the related work, Section 3 formally defines the EA problem, Section 4 introduces the details of proposed LLM-Align, Section 5 presents the experimental results, Section 6 concludes this work.

2. RELATED WORK

2.1 Embedding-based EA

Embedding-based knowledge graph (KG) alignment methods leverage models like TransE and Graph Neural Networks (GNNs) to learn embeddings for entities, which are then used to identify equivalent entities within vector spaces. Earlier methods primarily focused on utilizing the structural information of KGs for alignment. These include TransE-based approaches such as MTransE, IPTransE [23], and BootEA [10], as well as GNN-based methods like MuGNN [12], NAEA [24], RDGCN [25], and AliNet [13].

In these approaches, entity embeddings are learned by incorporating information about entities and their relationships. MTransE encodes the structural details of KGs in separate spaces before transitioning between them. TPTransE and BootEA are iterative alignment methods that expand seed alignments by incorporating newly discovered ones. MuGNN utilizes a multi-channel GNN to learn KG embeddings that are tailored for alignment tasks. NAEA enhances the TransE model by introducing a neighborhood-aware attentional representation for embedding learning. RDGCN employs a relation-aware dual-graph convolutional network, which integrates relation information through attentive interactions between a KG and its dual relation counterpart. AliNet, another GNN-based model, aggregates information from both direct and distant neighborhoods.

To achieve better alignment results, some methods incorporate entity attributes or names from KGs. For instance, JAPE [9] employs the Skip-Gram model to perform attribute embedding, capturing attribute correlations within KGs. GCN-Align [11] uses Graph Convolutional Networks (GCNs) to encode attribute

information into entity embeddings. MultiKE [14] adopts a framework that unifies the perspectives of entity names, relations, and attributes to learn embeddings for entity alignment. CEA [16] combines structural, semantic, and string-based features of entities, integrating them with dynamically assigned weights.

2.2 Language Model-based EA

With the successful application of Pre-trained Language Models (PLMs) across various tasks, some approaches have started leveraging PLMs to model the semantic information of entities in knowledge graph (KG) alignment tasks. For example, AttrGNN [15] employs BERT to encode the attribute features of entities, encoding each attribute and value separately and then using a graph attention network to compute a weighted average of these attributes and values. BERT-INT [26] uses a language model to embed the names, descriptions, attributes, and values of entities, and performs pair-wise neighbor-view and attribute-view interactions to compute the entity matching score. However, these interactions are time-consuming, limiting BERT-INT's scalability to larger KGs. SDEA [27] fine-tunes BERT to encode the attribute values of an entity into attribute embeddings, which are then processed by a BiGRU to obtain relation embeddings for the entity. TEA [28] organizes triples alphabetically by relations and attributes to form sequences and uses a textual entailment framework for entity alignment. TEA inputs entity-pair sequences into a PLM and has the PLM predict the probability of entailment. Like BERT-INT, TEA's pairwise input approach limits its scalability to large KGs. AutoAlign creates attribute character embeddings and predicate-proximity-graph embeddings using large language models.

With the emergence of Large Language Models (LLMs), several approaches have begun exploring their potential for entity alignment (EA). LLMEA [21] integrates knowledge from knowledge graphs (KGs) and large language models (LLMs) to predict entity alignments. Initially, it employs RAGAT to learn entity embeddings, which aid in identifying alignment candidates. These candidates are then transformed into multiple-choice questions for LLMs to determine the correct alignments. More specifically, LLMEA performs a multiple round of multi-choice QA, guilding the LLM to iteratively select the equivalent entity. ChatEA [22] first applies Simple-HHEA [29] to generate alignment candidates, after which it uses the reasoning capabilities of LLMs to predict the final alignments. LLMs are used to generate alignment scores of entities via in-context learning. Alignments are predicted by selecting entities with top scores. A rethinking process is also performed in ChatEA to enhance the EA results. LLM4EA [30] generates pseudo-labels for entity pairs using LLMs, followed by a probabilistic reasoning method to refine these labels. LLM4EA uses an LLM as alignment annotator to generate pseudo seed alignments, which are then used to train a GCN-based model. The final alignments are predicted by the GCN-based model. Seg-Align [31] first uses SDEA to obtain candidate alignments, then segments these candidates and selects the appropriate ones for LLM processing. Entity alignments are generated by prompting an LLM with multi-choice selection questions.

These LLM-based EA methods have demonstrated the potential of LLMs in EA tasks. They generally follow a two-step paradigm: first, they employ traditional EA models (e.g., RAGAT, Simple-HHEA, or SDEA) to generate candidate entity pairs; then, they rely on LLMs to make alignment decisions by treating

the task as a classification or ranking problem. However, the role of LLMs in these frameworks is often limited. For instance, LLMEA and Seg-Align formulates alignment as a multiple-choice question but uses fixed templates and does not actively filter informative triples. ChatEA employs LLMs for multi-hop reasoning but does not explicitly address challenges like prompt length, or irrelevant information. Our approach LLM-Align fully embraces the reasoning capabilities of LLMs by integrating a three-stage framework with heuristic-driven prompt construction and a multi-round voting mechanism. LLM-Align constructs informed prompt by selecting attributes and relations based on identifiability, ensuring that only the most informative information is passed to the LLM. LLM-Align also uses a multi-round voting mechanism to address hallucination and positional bias issues inherent in LLM generation.

3. PROBLEM DEFINITION

Knowledge Graph. Knowledge Graphs (KGs) represent the structural information of real-world entities through triples in the form $\langle s, p, o \rangle$. There are two kinds of triples in KGs, relational ones and attributive ones. Relational triples capture the relationships between entities, while attributive triples describe the attributes of entities. Formally, we represent a KG as $G = (E, R, A, L, T_{att}, T_{rel})$, where E, R, A, and L are sets of entities, relations, attributes, and literals, respectively, $T_{att} \subseteq (E \times A \times L)$ is the set of attributive triples, $T_{rel} \subseteq (E \times R \times E)$ is the set of relational triples.

Entity Alignment. Given two KGs, a source KG $G = (E, R, A, L, T_{att}, T_{rel})$ and target KG $G' = (E', R', A', L', T'_{att}, T'_{rel})$, the goal of KG alignment is to identify the equivalent target entity in E' for each source entity in E.

4. PROPOSED APPROACH

In this section, we introduce our proposed approach LLM-Align. Figure 1 shows the framework of LLM-Align, which works in three stages:

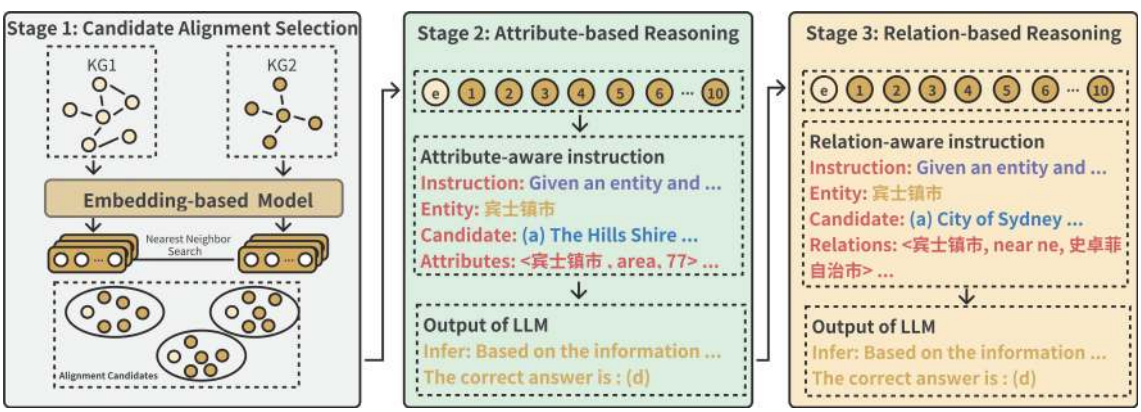

Figure 1. The framework of LLM-Align.

- **Candidate Alignment Selection.** This stage derives entity embeddings by using an existing embedding-based EA model. Subsequently, the nearest neighbor search is performed to obtain candidate alignments for each source entity.
- **Attribute-based Reasoning.** In this stage, LLM-Align identifies and extracts the most informative attributes of entities to construct the inputs for LLMs. A multi-round voting mechanism is employed to conduct multiple parallel reasoning. If a candidate entity is selected in a certain number of reasoning results, it is output as the aligned result; if no entity meets the criterion, LLM-Align moves to the next stage, relation-based reasoning.
- **Relation-based Reasoning.** This stage selects informative relations of entities to construct the inputs for LLMs. Similar to the previous stage, this stage also uses a heuristic relation selection strategy and employs the multi-round voting mechanism to get the alignment results.

In the following, we introduce our proposed approach in detail.

4.1 Candidate Alignment Selection

Given a set of source entities E and a set of target entities E', candidate alignment selection is to obtain a set of candidate target entities C_e for each source entity $e \in E$, where $C_e \subset E'$. More specifically, we employ an existing EA model which generates similarity scores for entity pairs. Target entities having the largest similarities with source entity will be selected as candidates. To control the size of input to LLMs, we let $|C_e| << |E'|$. To avoid introducing any prior biases to LLMs, the previous scores of entities will be ignored in the LLM-based EA reasoning stage.

4.2 EA Prompts for LLMs

Once a set of candidate alignments are obtained, they will be passed into an LLM to infer the final results. Here we format EA task as single choice selection problems and let LLMs to generate the alignment results based on the given prompts. Specifically, the prompt of an EA task contains three parts, including the instruction of EA task, the information of a source entity, and a list of candidate target entities. LLMs are asked to select the most likely target entity for the source entity. Figure 2 shows examples of EA prompts for LLMs.

Using different background information, we can construct three types of prompts: (1) Knowledge-driven Prompts: prompts with only entity names; (2) Attribute-aware Prompts: prompts with entity attributes; (3) Relation-aware Prompts: prompts with entity relations.

Knowledge-driven Prompts. Knowledge-driven prompts only provide LLMs the names of entities, LLMs have to infer the results based on their own knowledge about these entities. For example, as shown in Figure 2, the knowledge-driven prompt uses entity names as questions and options. The correct alignment is (*City of Bankstown*, 宾士镇市), the LLMs are supposed to select the correct target entity *City of Bankstown* from the candidate list.

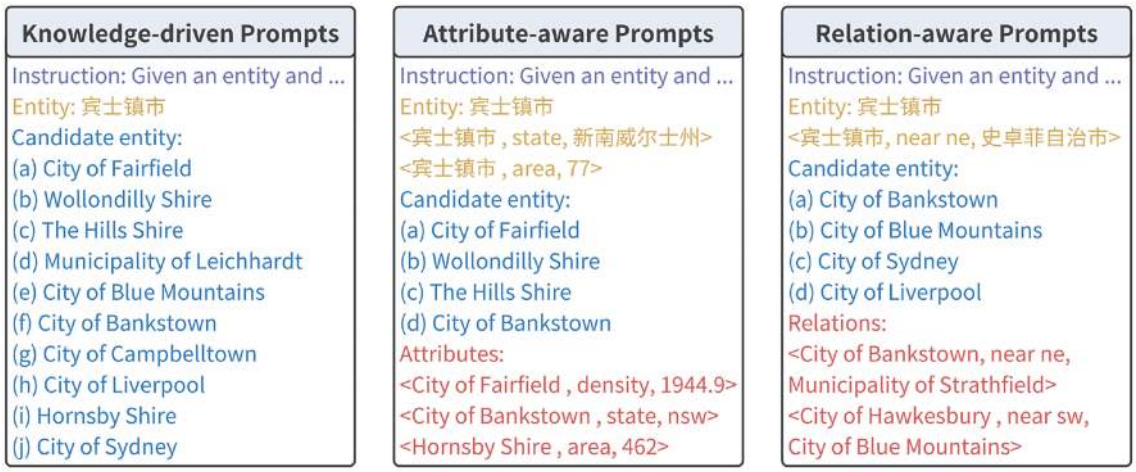

Figure 2. Prompts for LLM-Align.

Attribute-aware Prompts. Attribute-aware prompts provide entities' attribute information on the basis of knowledge-driven prompts. It is believed that Entities' attribute information is helpful for LLMs to infer the correct alignment. As shown in Figure 2, the attribute information of both source entity and candidate target entities are provided with triples. For example, the state and area of 宾士镇市 are given in the prompt, and the population density of *City of Fairfield*, the state of *City of Bankstown*, and the area of *Hornsby Shire* are also provided in the prompt.

Relation-aware Prompts. Considering attributes of entities might not provide sufficient information to find entity alignments, relation-aware prompts include relation triples of entities to provide evidence for finding alignments. As shown in Figure 2, the relation triples identifying nearby cities of source and candidate target entities are given in the prompt.

To get accurate alignment results, LLM-Align uses attribute-aware prompts and relation-aware prompts in sequence. Attribute-aware prompts are applied first, and if no alignment is reached via majority voting, relation-aware prompts are used. The final alignment is taken from the most confident stage rather than aggregating results across stages.

4.3 Heuristic Attribute and Relation Selection

In the stages of Attribute-based and Relation-based EA reasoning, LLM-Align identifies and extracts the most informative attributes and relations of entities to construct attribute-aware prompts and relation-aware prompts for LLMs. In this work, we define heuristic rules to guide the selection of attributes and relations.

Before introducing the heuristic rules, we first define the **function** and **functionality** of a predicate (either a relation or an attribute) in triples. A predicate p is considered a function if, for a given subject

entity, there is exactly one object associated with p. For example, the relation *birthPlace* and the attribute *age* are functions because a specific person has only one birthplace and one age. Such predicates are particularly useful for uniquely identifying entities.

In the work of PARIS [32], **functionality** is defined to measure to what extent a predicate is a function one, which is estimated by

$$fun(p) = \frac{\left|\{s \mid (s, p, o) \in T\}\right|}{\left|\{(s, o) \mid (s, p, o) \in T\}\right|} \tag{1}$$

where T is the set of triples in a KG. The higher the functionality, the more important the predicate is for uniquely identifying entities.

In this work, we extend the functionality measure to the context of two KGs, and define the **attribute functionality** $fun_{att}(a)$ and **relation functionality** $fun_{rel}(r)$. Let source KG be $G = (E, R, A, L, T_{att}, T_{rel})$ and target KG be $G' = (E', R', A', L', T'_{att}, T'_{rel})$, $fun_{att}(a)$ and $fun_{rel}(r)$ are computed as:

$$fun_{att}(a) = \frac{|\{h \mid (h, a, v) \in \{T_{att} \cup T'_{att}\}\}|}{|\{(h, v) \mid (h, a, v) \in \{T_{att} \cup T'_{att}\}\}|} \tag{2}$$

$$fun_{rel}(r) = \frac{|\{h \mid (h, r, t) \in \{T_{rel} \cup T'_{rel}\}\}|}{|\{(h, t) \mid (h, r, t) \in \{T_{rel} \cup T'_{rel}\}\}|} \tag{3}$$

The functionality of an attribute or relation reflects its importance in identifying entities, and can therefore guide the selection of informative attributes and relations for entity matching.

4.3.1 Heuristic Attribute Selection

For a source entity $e \in E$, a set of its candidate target entities $C_e \subset E'$ is obtained in the candidate alignment selection stage. To extract informative attributes, we define a metric called *identifiability*, which quantifies the significance of an attribute in distinguishing between different entities. A higher *identifiability* score indicates a more important attribute. Given an attribute a, we use $identy_{att}(a)$ to denote the *identifiability* of it. $identy_{att}(a)$ is computed based on the attribute functionality $fun_{att}(a)$ and the frequency $freq_{att}(a)$ of attribute a.

The frequency $freq_{att}(a)$ of attribute a in the triples of candidate target entities C_e is computed as:

$$freq_{att}(a, C_e) = \frac{|\{h \mid h \in C_e \wedge (h, a, v) \in T'_{att}|}{|C_e|} \tag{4}$$

The *identifiability* is a combination of the attribute functionality $fun_{att}(a)$ and the frequency $freq_{att}(a)$, which is computed as:

$$identy_{att}(a, C_e) = fun_{att}(a) \times freq_{att}(a, C_e) \tag{5}$$

For each entity in $\{e\} \cup C_e$, we first compute the metrics of *identifiability* for all the attributes of the entity. Then the top-k attributes with the highest *identifiability* are selected, triples of these attributes are added to the attribute-aware prompts for LLMs to decide the true target entity for e.

4.3.2 Heuristic Relation Selection

Similar to the attribute selection process, we also compute the *identifiability* metrics for relations, and then select the top-k relations to present in relation-aware prompts. For a relation r, its frequency $freq_{rel}(r, C_e)$ is computed as:

$$freq_{rel}(r, C_e) = \frac{|\{h \mid h \in C_e \wedge (h, r, t) \in T'_{rel}\}|}{|C_e|} \tag{6}$$

The *identifiability* of relation r is computed as:

$$identy_{rel}(r, C_e) = fun_{rel}(r) \times freq_{rel}(r, C_e) \tag{7}$$

4.3.3 Discussion on Heuristic Rule Design

In this work, we select attributes and relations based on their functionality and frequency, aiming to identify those most informative for distinguishing between entities. The proposed metric, termed identifiability, is defined as the product of functionality and frequency. This design is grounded in the intuition that predicates which are both function-like (i.e., exhibit one-to-one mappings) and frequently appear across candidate entities are more useful for entity alignment.

Functionality, as introduced in PARIS [32], measures how close a predicate is to being a function—i.e., how uniquely it maps a subject to an object. In the context of entity alignment, high-functionality predicates (e.g., *birthPlace*, *age*) help uniquely identify an entity and thus improve the quality of alignment decisions. Frequency captures the relevance of a predicate in the specific alignment context, ensuring that selected predicates are not only generally useful but also applicable to the current candidate set. By combining these two aspects, the identifiability metric reflects both global distinctiveness and local applicability. This dual consideration balances between universal importance and task-specific effectiveness, making it suitable for prompt construction in LLM-based reasoning.

Other heuristic strategies, such as attention-based methods, can dynamically learn the importance of predicates based on alignment outcomes. However, these approaches typically introduce additional computational overhead and require labeled supervision. We plan to explore such alternative heuristic rules in future work.

4.4 Multi-round Voting Mechanism

When processing long texts, LLMs are affected by positional bias, meaning their performance varies significantly depending on the position of the information within the text. Experiments show that LLMs

handle information at the beginning and end of documents more effectively compared to information in the middle. Additionally, hallucinations remain a challenge in practical applications. Factors like model size, training data quality, and the training process contribute to hallucinations. To mitigate the effects of hallucination and positional bias in EA reasoning, we propose a multi-round voting mechanism.

The multi-round voting mechanism performs multiple independent reasoning. Let the number of votes be n, and the size of the candidate entity set be m. The method first samples n unique permutations from the $m!$ possible arrangements of the candidate set. The large language model then performs parallel reasoning on the n inputs to generate n independent outputs, and the final answer is determined through a voting mechanism. Let $count(c)$ be the number of times that a target entity c appears in the LLM reasoning results. The target entity c^* with the highest $count(c^*) \geq \lfloor n / 2 \rfloor$ will be chosen as the final alignment result for the source entity. If there is no target entity satisfying $count(c) \geq \lfloor n / 2 \rfloor$, the multi-round voting will not output any alignment results.

The multi-round voting mechanism described above effectively mitigates the impact of positional bias and hallucinations on large language models, enhancing their accuracy and stability in entity alignment reasoning tasks. Additionally, the multi-round voting mechanism is similar to the concept of ensemble learning, implicitly integrating multiple models, which improves the model's ability to solve complex and difficult problems. In the subsequent experiments, the paper validates the effectiveness of the multi-round voting mechanism in entity alignment tasks, showing that compared to single-round reasoning, the multi-round voting mechanism provides more stable performance in alignment reasoning.

5. EXPERIMENTS

5.1 Experiment Settings

5.1.1 Datasets

In our experiments, we utilize the DBP15K datasets, created by Sun et al. [9]. These datasets are derived from DBpedia, a large-scale multilingual knowledge graph that contains abundant inter-language links between different language versions. Specific subsets of the Chinese, English, Japanese, and French versions of DBpedia were selected according to certain criteria. Using these subsets of DBpedia, three cross-lingual EA datasets were built, including Chinese-English (ZH-EN), Japanese-English (JA-EN), and French-English (FR-EN). Details of these datasets are shown in Table 1.

5.1.2 Models Settings

EA Models for Candidate Alignment Selection. In the experiments, we use the DERA-R [33] and GCN-Align [11] models as the base models for candidate entity selection. The DERA-R method and GCN-Align model respectively use textual information and structural information for modeling, both achieving good performance on the Hits@10 metric. Conducting diverse experiments with these two

Table 1. Statistics of DBP15K Datasets.

Dataset	Lang.	Entity	Rel.	Attr.	Rel.triples	Attr.triples
ZH-EN	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
JP-EN	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
FR-EN	French	66,858	1,379	4,547	192,191	528,665
	English	105,889	2,209	6,422	278,590	576,543

candidate alignment selection models aims to validate the generality of the proposed method, while also allowing this study to further explore how the candidate entity sets retrieved by different information-based modeling approaches impact the final alignment reasoning.

LLMs for EA Reasoning. The experiment uses Qwen1.5-32B-Chat [34] and Qwen1.5-14B-Chat [34] as reasoning models. Both models can perform efficient reasoning using the vLLM framework on a single 80G GPU. Conducting experiments with models of different scales helps us explore the performance of the method under varying computational resources. At the same time, models of different sizes allow this study to investigate the impact of model scale on reasoning alignment tasks.

5.1.3 Baselines

To comprehensively evaluate the performance of this method, we use the following EA models for comparison:

- DERA [33]: The EA method based on heterogeneous parsing with large language models. This method achieved state-of-the-art (SOTA) performance across multiple datasets.
- DERA-R [33]: A simplified version of the DERA method, which only uses the heterogeneous parsing module and candidate entity retrieval module, without introducing the related sequential re-ranking module.
- TEA [28]: A classic method based on language model modeling, included for comparison to explore the impact and effectiveness of large language models and pre-trained language models on the entity alignment task.
- BERT-INT [26]: It is a EA method based on BERT language model, it performs cross-graph interactive modeling of semantic information such as entity names, relationships, and attributes.
- HMAN [35]: This method uses fine-grained ranking techniques from traditional information retrieval in the final stage to re-rank the candidate entity set.

- AttrGNN [15]: It is a representative method for modeling the topological structure of attribute triples. By comparing with this model, this study explores the impact of using attribute information modeling through graph neural networks and large language models on the entity alignment task.
- LLMEA [21] integrate knowledge from KGs and LLMs to predict entity alignments.
- ChatEA [22] first use an existing EA model to generate alignment candidates and then leverages the reasoning capabilities of LLMs to predict the final results.
- Seg-Align [31] uses SDEA to get candidate alignments, and then segments candidates, selecting suitable ones for processing by LLMs.

5.2 Overall Results

Table 2 shows the overall results on DBP15K datasets. LLM-Align employs GCN-Align and DERA-R as base models for candidate alignment selection, while Qwen1.5-14B-Chat and Qwen1.5-32B-Chat serve as reasoning models. LLM-Align adopts a multi-round voting mechanism to make final alignment decisions. For each source entity, it selects a single target entity that satisfies the voting criterion, without generating similarity scores or ranking a list of candidates. Therefore, unlike traditional EA methods that output a ranked list of candidates, LLM-Align inherently produces only one aligned entity per source and cannot compute standard ranking-based metrics such as Hits@10. Therefore, only Hits@1 metrics for LLM-Align are shown in Table 2. The GCN-Align and DERA-R results were reproduced by us, while results of other methods are sourced from their original papers.

Experimental results demonstrate that LLM-Align is highly effective, achieving state-of-the-art performance. In combination with GCN-Align, LLM-Align significantly boosts results: with Qwen1.5-14B-Chat as the base LLM, it increases Hits@1 by 32.9%, 34.0%, and 37.3% on the ZH-EN, JA-EN, and FR-EN datasets, respectively. When using Qwen1.5-32B-Chat, LLM-Align achieves even greater improvements of 34.9%, 34.7%, and 38.0% in Hits@1 across the same datasets.

When using DERA-R and Qwen1.5-14B-Chat, LLM-Align achieves Hits@1 scores of 97.8%, 95.7%, and 99.2% on the ZH-EN, JA-EN, and FR-EN datasets, respectively. With the larger model, Qwen1.5-32B-Chat, performance improves further, reaching 98.3% on ZH-EN, 97.6% on JA-EN, and 99.5% on FR-EN. Compared to the base model DERA-R, LLM-Align effectively boosts Hits@1 scores with both 14B and 32B models, yielding increases of 2.3%, 0.7%, and 0.1% in Hits@1 with the 14B LLM, and gains of 3.2%, 2.6%, and 0.4% with the 32B LLM. Among all baselines, LLM-Align with the 32B model achieves the highest Hits@1 across all three datasets.

While LLM-Align achieves the best overall performance when combined with a strong candidate selector like DERA-R and a powerful LLM (Qwen-32B), it is important to emphasize that our framework is not limited to high-performing base models. As shown in Table 2, when using GCN-Align—an earlier and weaker EA model—as the candidate generator, LLM-Align still brings substantial performance gains. For instance, Hits@1 on the ZH-EN dataset increases from 0.420 (GCN-Align) to 0.749 with LLM-Align (Qwen-14B), and further to 0.769 with Qwen-32B, clearly demonstrating the effectiveness of LLM-Align in leveraging LLM reasoning across different baseline strengths.

Table 2. Overall Results on DBP15K Datasets.

Model	ZH-EN		JA-EN		FR-EN	
	Hits@1	Hits@10	Hits@1	Hits@10	Hits@1	Hits@10
GCN-Align	0.420	0.790	0.445	0.815	0.432	0.812
TEA	0.941	0.983	0.941	0.979	0.987	0.996
BERT-INT	0.968	0.990	0.964	0.991	0.992	0.998
HMAN	0.871	0.987	0.935	0.994	0.973	0.998
AttrGNN	0.796	0.929	0.783	0.921	0.919	0.978
DERA	0.968	0.994	0.967	0.992	0.989	0.999
DERA-R	0.955	0.992	0.950	0.989	0.991	1.000
LLMEA	0.898	0.923	0.911	0.946	0.957	0.977
ChatEA	-	-	-	-	0.990	1.000
Seg-Align	0.953	-	0.907	-	0.987	-
LLM-Align(GCN-Align-Qwen14B)	0.749	-	0.785	-	0.805	-
LLM-Align(GCN-Align-Qwen32B)	0.769	-	0.792	-	0.812	-
LLM-Align(DERA-R-Qwen14B)	0.978	-	0.957	-	0.992	-
LLM-Align(DERA-R-Qwen32B)	**0.983**	-	**0.976**	-	**0.995**	-

5.3 Ablation Study

In this paper, we conducted a series of ablation experiments on base models of different scales (14B and 32B) to gain deeper insights into the effectiveness and significance of each module within the framework. The ablation studies focused on the Attribute-based Reasoning (**AR**), Relation-based Reasoning (**RR**), and Multi-round Voting (**MV**) components. Table 3 shows the results of ablation study.

LLM-Align without AR Module. In ablation experiments with the 14B model, the Hits@1 metric dropped by 16.1%, 15.3%, and 14.0% on the three datasets, respectively. This result demonstrates that, when performing reasoning alignment with a smaller model, the absence of the Attribute-based Reasoning module significantly degrades EA performance, underscoring its importance within the framework. With the 32B model, Hits@1 decreased by 11.1%, 11.1%, and 2.2% across the three datasets. Although the reduction was less severe compared to the 14B model, removing the Attribute-based Reasoning module still had a substantial negative effect on reasoning alignment performance relative to the full framework.

Table 3. Results of Ablation Study.

AR	RR	MV	ZH-EN		JA-EN		FR-EN	
			14B	32B	14B	32B	14B	32B
✓	✓	✓	**0.978**	**0.983**	**0.957**	**0.976**	**0.992**	**0.995**
✗	✓	✓	0.817	0.872	0.804	0.865	0.852	0.973
✓	✗	✓	0.952	0.980	0.938	0.973	0.990	0.994
✓	✓	✗	0.918	0.964	0.926	0.968	0.954	0.986
✗	✓	✗	0.731	0.813	0.722	0.823	0.794	0.933
✓	✗	✗	0.909	0.971	0.914	0.952	0.947	0.981

LLM-Align without RR Module. After removing the Relation-based Reasoning module from the complete reasoning framework, experimental results showed a performance drop in both the 14B and 32B models, though the degree of decline varied significantly. In the 14B model, performance decreased by an average of 15.7% across the three datasets, whereas in the 32B model, performance remained nearly stable, with only a slight decline of 1%-3%.

LLM-Align without MV Module. The Multi-round Voting module was designed to address issues such as hallucinations and positional bias that can occur during large language model generation. By shuffling option order and requiring an answer to receive a majority of votes across multiple rounds, it enhances the model's confidence in aligning the correct entities. After removing this module, performance on the 14B and 32B models dropped by an average of 4.3% and 1.2%, respectively, highlighting its effectiveness within the reasoning alignment framework. Analysis of the generation results revealed that large language models often make errors in single-round reasoning decisions. However, with the Multi-round Voting module, the model can still arrive at the correct answer even if a mistake is made in one round of reasoning. This insight offers valuable guidance for designing future entity alignment frameworks based on large language models, especially in balancing model size with accuracy.

LLM-Align with Only RR Module. In this experiment, both the Attribute-based Reasoning and Multi-round Voting modules were removed, leaving only the Relation-based Reasoning module. Under this setup, Hits@1 performance on the 14B and 32B models dropped by an average of 22.7% and 12.8%, respectively, compared to the complete reasoning alignment frame-work. Reintroducing the Attribute-based Reasoning and Multi-round Voting modules led to a marked improvement in reasoning alignment performance, with optimal results achieved when all modules were used together. This experiment demonstrates that the modules complement one another, collectively enhancing the effectiveness of reasoning alignment.

LLM-Align with Only AR Module. In this setting, both the Relation-based Reasoning and Multi-round Voting modules were removed. Results showed that with the 14B model, average Hits@1 performance reached 92.3%, while with the 32B model, it reached 96.8%.

The above experiment results show that the performance of LLM-Align improved as RR and MV modules were incrementally added with AR module. Based on these results, it is inferred that the synergistic effect among modules stems from the complementarity of information and enhanced confidence in the information. Adding the Relation-based Reasoning module to the Attribute-based Reasoning module complements attribute information with structural context, while the Multi-round Voting module strengthens confidence in valuable information. Thus, when all three modules operate together, the framework achieves higher performance than with any one or two modules alone.

5.4 Analysis on the Impact of Candidate Alignment Orders

In this subsection, we examine how the order of candidate entities in prompts impacts the final alignment results. Through experiments with different orders, we analyze whether these variations significantly affect EA outcomes. Specifically, we test the following orders: (1) ordered by similarities, arranged in descending similarity score as determined by the candidate selection model; (2) random order, where candidate entities are randomly shuffled; and (3) reverse order, which is the inversion of the original order.

This experiment was conducted using knowledge-driven prompts and attribute-aware prompts. Figure 3 illustrates the results of different entity orders. The findings indicate that for both the 14B and 32B base reasoning models, retaining the original order from the candidate alignment selection process yielded the best results, followed by random order, while reverse order performed the worst. This suggests that the order information generated during the candidate alignment selection process may contain implicit cues that aid the model's reasoning. The order information reflects the position of the correct answer in the candidate list, highlighting that this position significantly impacts final performance.

When the correct answer is positioned at the front of the list, the likelihood of the model directly identifying it increases due to the order information. For models like DERA-R, which maintain high Hits@1 performance during the candidate matching process, keeping the original order enhances the chances of the correct answer appearing at the top of the candidate list, potentially further boosting the model's performance. Conversely, if the correct entity is located toward the back of the list, this ordering may negatively affect performance by diminishing the model's ability to locate the correct entity. This effect is particularly pronounced for candidate matching models with low Hits@1 performance, as retaining the original order could place the correct answer at the end of the candidate list, thus reducing overall effectiveness. The experiments above demonstrate that positions significantly influence the final EA results of LLMs, with these models showing a preference for selecting candidates ranked at the top of the list.

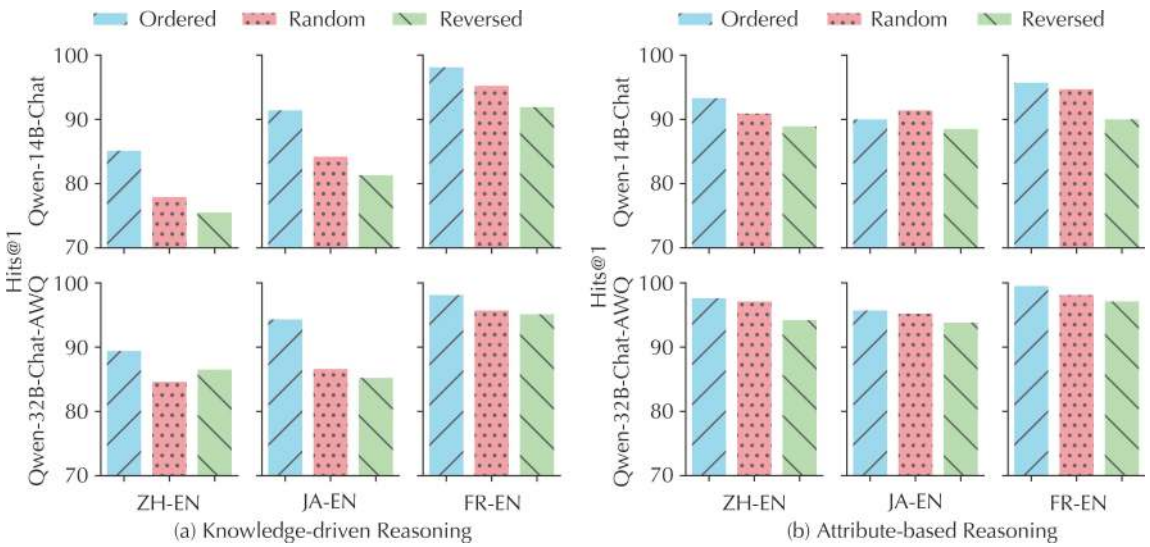

Figure 3. Experimental Results Affected by Positional Bias.

5.5 Analysis on the Impact of LLM Size

To investigate the impact of model size on the alignment effectiveness of the proposed method, we conducted experiments using LLMs of varying sizes in the EA inference, specifically 1.5B, 14B, and 32B. The DERA-R model was employed to generate the candidate alignments, with the size of the candidate alignments set to 10. The experimental results are presented in Figure 4.

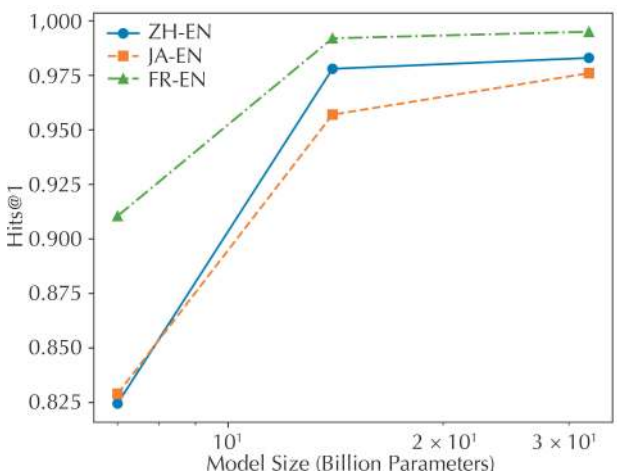

Figure 4. Impact of Model Scale on EA Reasoning.

The experimental results indicate that as the model size increases from 1.5B to 32B, performance across all three datasets exhibits a gradual upward trend. This finding reinforces the notion that there is a positive correlation between model size and reasoning alignment performance, suggesting that larger language models can achieve higher Hits@1 scores within the reasoning alignment framework proposed in this study.

Notably, the accuracy of the 1.5B model hovers around 9% across all datasets, which is close to random selection performance given that the candidate entity set size is 10. Analysis of the model's outputs reveals that the 1.5B model has limited instruction-following capabilities and struggles to accurately understand and execute the reasoning alignment task. This suggests that within the framework proposed in this study, there is a lower limit on model size; below this threshold, the model may be ineffective in performing the reasoning alignment task.

The paper further explores how model size affects the handling of entities with varying difficulty. To precisely differentiate entity difficulty, a grouping method based on similarity calculations during the candidate matching phase was implemented. Specifically, the DERA-R model was first used to match candidate entities for the test set, selecting the top 10 candidates. An entity's difficulty level was determined by whether DERA-R ranked the correct entity first: if the correct entity was not ranked first, it was classified as a high-difficulty entity; otherwise, it was categorized as a low-difficulty entity.

To ensure the stability and reliability of the results, 300 samples were randomly selected from both the high-difficulty entity set and the low-difficulty entity set, and the experiments were repeated three times on these samples. The average result was taken as the final outcome. The experimental results are shown in Figure 5. The results indicate that as the model size increases, the Hits@1 performance improvement for the high-difficulty entity set is greater than for the low-difficulty entity set. These findings align with intuition, as larger models are better equipped to handle more complex and challenging entities.

Although expanding the model size significantly improves its ability to handle high-difficulty entities, we observed that overall performance on the low-difficulty entity set remains higher than on the high-difficulty set. This phenomenon reveals consistency between the reasoning alignment process and the candidate matching process, indicating that even when the candidate matching process has already correctly identified the aligned entity, the reasoning process can maintain or even improve performance.

This consistency suggests that for entities already ranked first in the candidate matching process, the reasoning alignment process can further enhance the accuracy without compromising the original performance. However, for entities not ranked first during the candidate matching process, although the reasoning alignment process faces certain challenges, it often resolves some of the uncertainties left by the candidate matching process, thus substantially improving overall performance.

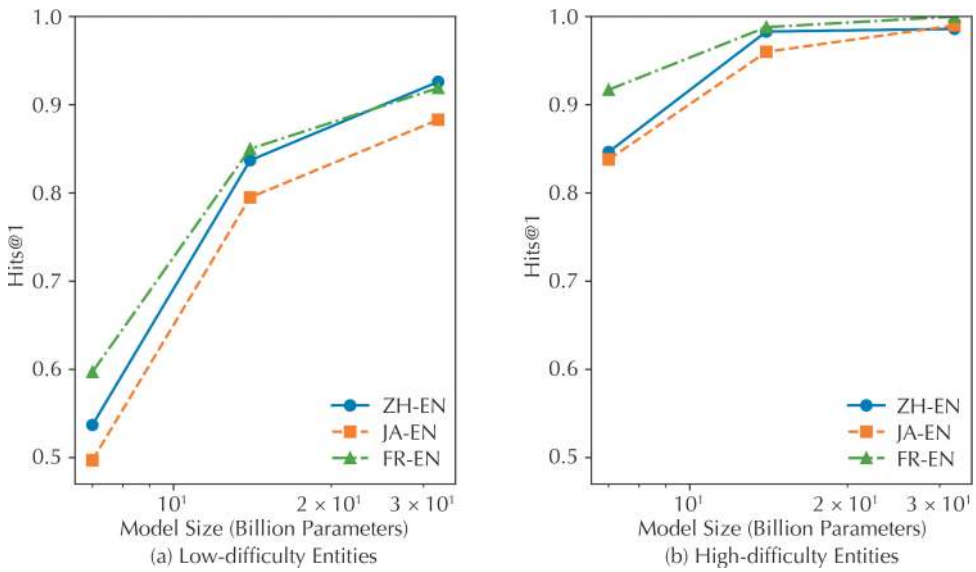

Figure 5. Impact of Model Scale on EA for high-difficulty entity and low-difficulty entity.

5.6 Analysis on the Impact of Candidate Size

This section aims to explore the ability of the reasoning alignment framework to handle candidate entity sets of varying sizes. By adjusting the size of the candidate entity set, five different scales were used: 10, 20, 30, 40, and 50. The goal was to observe how the accuracy of the proposed framework changes when faced with different numbers of candidate entities. These experiments were conducted on two large language models with sizes of 14B and 32B, using the DBP15K dataset across three datasets, to obtain comprehensive experimental results.

In this experiment, we removed all three modules from the reasoning alignment framework, using only knowledge-driven prompts for LLMs. This setup allows for an accurate evaluation of how the size of the candidate entity set impacts the final results, free from the influence of individual or combined optimizations of the framework's specific modules. To ensure stability and reliability, we extracted 500 samples from the test set, ensuring that the correct entity was included in the top-k candidate entities. The values of top-k were set at 10, 20, 30, 40, and 50, and three sampling experiments were conducted for each candidate set size, with the final result being the average of these three experiments. Natural language instructions used were based on the knowledge-driven instructions proposed in the second section of this chapter.

The experimental results, shown in Figure 6, were consistent for both the 14B and 32B models. As the number of candidate entities increased, Hits@1 performance showed a downward trend. This trend aligns with expectations: as the number of candidate entities grows, the large language model must make judgments and inferences over more options. This can cause the model's attention to be dispersed across more irrelevant entity options, thereby affecting its ability to correctly identify the target entity.

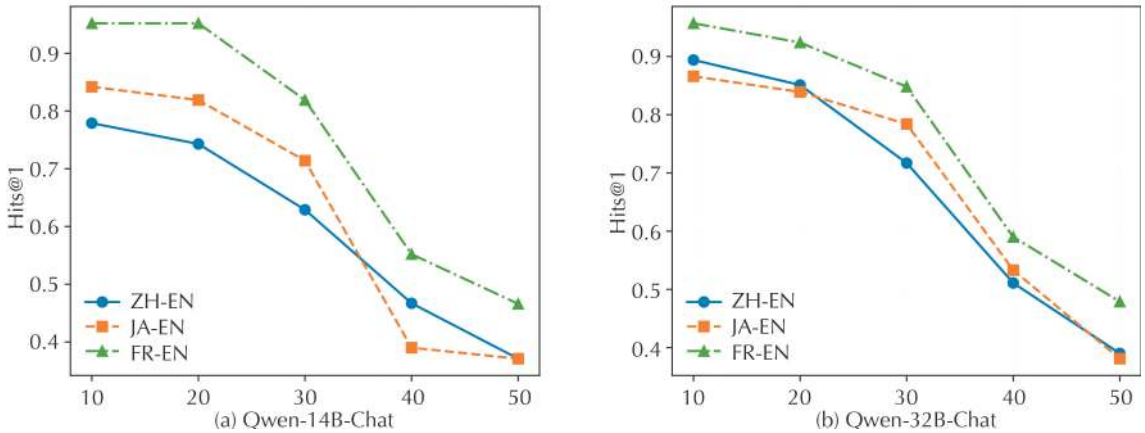

Figure 6. The impact of the number of candidate entities on the Hits@1 metric.

Overall, the 32B model demonstrated superior performance in this experiment, further validating the importance of model size in improving reasoning alignment performance. However, it is noteworthy that in the FR-EN dataset, when the number of candidate entities was 20, 40, or 50, the performance of the 32B model was on par with, or even slightly lower than, that of the 14B model. Upon analyzing the incorrect samples, we found that most of these errors involved entities with similar names. The 14B model tended to directly output the similar entity as the correct answer, while the 32B model, relying on its own knowledge, attempted to analyze and infer between these similar entities. However, during this inference process, issues arose, leading the model to ultimately arrive at the wrong answer.

The experimental results, as shown in Figure 6 indicate that for both 14B and 32B scale models, the Hits@1 performance decreases as the number of candidate entities increases. This trend is expected, as the large language model has to reason about more options, potentially diverting attention to more unrelated entities, which affects the judgment of the correct entity. Overall, the 32B scale model performed better in this experiment, further validating the importance of model scale in improving inference alignment performance. It is noteworthy that on the FR-EN dataset, the performance of the 32B scale model is equal to or slightly lower than that of the 14B scale model when the number of candidate entities is 20, 40, and 50. The erroneous samples revealed that most of these errors are due to entities with similar names. The 14B model directly output the similar entity as the correct answer, while the 32B model attempted to analyze and infer based on its own knowledge, resulting in incorrect answers.

6. CONCLUSIONS

In this work, we propose an entity alignment approach based on large language models, termed LLM-Align. LLM-Align constructs EA prompts that include candidate alignments for LLMs, leveraging their reasoning capabilities to produce final alignment results. The method employs heuristic techniques to identify key attributes and relationships of entities, incorporating the selected triples into prompts for

the LLMs. To ensure high-quality alignment, we design a multi-round voting mechanism that mitigates position bias and addresses hallucination issues associated with LLMs. Experiments conducted on three EA datasets demonstrate that our approach effectively enhances the EA results of existing models. Compared to baselines, LLM-Align achieves the best performance when paired with a robust EA model for candidate selection.

AUTHOR CONTRIBUTIONS

Xuan Chen: proposed the research problems; designed the research framework; performed the research; collected and analyzed the data. Tong Lu: wrote and revised the manuscript. Zhichun Wang: proposed the research problems; designed the research framework; wrote and revised the manuscript.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 62276026).

REFERENCES

[1] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, "Knowledge graphs: Opportunities and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, p. 13071–13102, Apr. 2023. [Online]. Available: https://doi.org/10.1007/s10462–023–10465–9

[2] R. Reinanda, E. Meij, and M. de Rijke, "Knowledge graphs: An information retrieval perspective," *Found. Trends Inf. Retr.*, vol. 14, no. 4, p. 289–444, Oct. 2020. [Online]. Available: https://doi.org/10.1561/1500000063

[3] G. Balloccu, L. Boratto, G. Fenu, and M. Marras, "Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 646–656. [Online]. Available: https://doi.org/10.1145/3477495.3532041

[4] R. Sun, X. Cao, Y. Zhao, J. Wan, K. Zhou, F. Zhang, Z. Wang, and K. Zheng, "Multi-modal knowledge graphs for recommender systems," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1405–1414. [Online]. Available: https://doi.org/10.1145/3340531.3411947

[5] E. Raisi and S. H. Bach, "Selecting auxiliary data using knowledge graphs for image classification with limited labels," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4026–4031.

[6] X. Pan, T. Ye, D. Han, S. Song, and G. Huang, "Contrastive language-image pre-training with knowledge graphs," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of Advances in neural information processing systems (NIPS2013)*, 2013, pp. 2787–2795.

[8] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1511–1517. [Online]. Available: https://doi.org/10.24963/ijcai.2017/209

[9] Z. Sun, W. Hu, and C. Li, "Cross-lingual entity alignment via joint attribute-preserving embedding," in *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16*. Springer, 2017, pp. 628–644.

[10] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4396–4402. [Online]. Available: https://doi.org/10.24963/ijcai.2018/611

[11] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, "Cross-lingual knowledge graph alignment via graph convolutional networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 349–357. [Online]. Available: https://aclanthology.org/D18–1032

[12] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li, and T.-S. Chua, "Multi-channel graph neural network for entity alignment," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1452–1461. [Online]. Available: https://aclanthology.org/P19–1140

[13] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu, "Knowledge graph alignment network with gated multi-hop neighborhood aggregation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 222–229, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5354

[14] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, and Y. Qu, "Multi-view knowledge graph embedding for entity alignment," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 5429–5435. [Online]. Available: https://doi.org/10.24963/ijcai.2019/754

[15] Z. Liu, Y. Cao, L. Pan, J. Li, Z. Liu, and T.-S. Chua, "Exploring and evaluating attributes, values, and structures for entity alignment," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 6355–6364. [Online]. Available: https://aclanthology.org/2020.emnlp-main.515

[16] W. Zeng, X. Zhao, J. Tang, and X. Lin, "Collective entity alignment via adaptive features," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 1870–1873.

[17] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang, "Generate rather than retrieve: Large language models are strong context generators," 2023. [Online]. Available: https://arxiv.org/abs/2209.10063

[18] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *Transactions on Machine Learning Research*, 2023.

[19] L. Pan, A. Albalak, X. Wang, and W. Y. Wang, "Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning," *arXiv preprint arXiv:2305.12295*, 2023.

[20] R. Zhang, Y. Su, B. D. Trisedya, X. Zhao, M. Yang, H. Cheng, and J. Qi, "Autoalign: Fully automatic and effective knowledge graph alignment enabled by large language models," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.

[21] L. Yang, H. Chen, X.Wang, J. Yang, F.-Y.Wang, and H. Liu, "Two heads are better than one: Integrating knowledge from knowledge graphs and large language models for entity alignment," 2024.

[22] X. Jiang, Y. Shen, Z. Shi, C. Xu, W. Li, Z. Li, J. Guo, H. Shen, and Y. Wang, "Unlocking the power of large language models for entity alignment," 2024.

[23] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4258–4264. [Online]. Available: https://doi.org/10.24963/ijcai.2017/595

[24] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "Neighborhood-aware attentional representation for multilingual knowledge graphs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 1943–1949. [Online]. Available: https://doi.org/10.24963/ijcai.2019/269

[25] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao, "Relation-aware entity alignment for heterogeneous knowledge graphs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 5278–5284. [Online]. Available: https://doi.org/10.24963/ijcai.2019/733

[26] X. Tang, J. Zhang, B. Chen, Y. Yang, H. Chen, and C. Li, "BERT-INT: a bert-based interaction model for knowledge graph alignment," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3174–3180, main track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/439

[27] Z. Zhong, M. Zhang, J. Fan, and C. Dou, "Semantics driven embedding learning for effective entity alignment," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 2127–2140.

[28] Y. Zhao, Y. Wu, X. Cai, Y. Zhang, H. Zhang, and X. Yuan, "From alignment to entailment: A unified textual entailment framework for entity alignment," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 8795–8806. [Online]. Available: https://aclanthology.org/2023.findings-acl.559

[29] X. Jiang, C. Xu, Y. Shen, Y. Wang, F. Su, Z. Shi, F. Sun, Z. Li, J. Guo, and H. Shen, "Toward practical entity alignment method design: Insights from new highly heterogeneous knowledge graph datasets," in *Proceedings of the ACM on Web Conference 2024*, ser.WWW'24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 2325–2336. [Online]. Available: https://doi.org/10.1145/3589334.3645720

[30] S. Chen, Q. Zhang, J. Dong, W. Hua, Q. Li, and X. Huang, "Entity alignment with noisy annotations from large language models," in *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: https://arxiv.org/abs/2405.16806

[31] L. Yang, J. Cheng, and F. Zhang, "Advancing cross-lingual entity alignment with large language models: Tailored sample segmentation and zero-shot prompts," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 8122–8138. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.475/

[32] F. M. Suchanek, S. Abiteboul, and P. Senellart, "Paris: probabilistic alignment of relations, instances, and schema," *Proc. VLDB Endow.*, vol. 5, no. 3, pp. 157–168, nov 2011. [Online]. Available: https://doi.org/10.14778/2078331.2078332

[33] Z. Wang and X. Chen, "Dera: Dense entity retrieval for entity alignment in knowledge graphs," 2024.

[34] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, et al. "Qwen technical report," 2023.

[35] H.-W. Yang, Y. Zou, P. Shi, W. Lu, J. Lin, and 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 (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4431–4441. [Online]. Available: https://aclanthology.org/D19–1451

AUTHOR BIOGRAPHY

Xuan Chen received his master degree from Beijing Normal University in 2024. He received his bachelor degree from China University of Geosciences (Beijing) in 2021. His research interests include knowledge graph, and language models.

Tong Lu is currently a Ph.D. candidate at Beijing Normal University. He earned his bachelor's degree from Hebei GEO University and his master's degree from Yunnan University. His research interests include knowledge graphs and large language models.

Zhichun Wang is currently an associate professor at the School of Artificial Intelligence, Beijing Normal University. He received his Ph.D. degree from Tianjin University in 2010, and worked as post-doctoral researcher in Tsinghua University from 2011 to 2012. His research interests include large language models and knowledge graphs.