EasyEA: Large Language Model is All You Need in Entity Alignment Between Knowledge Graphs

Jingwei Cheng†*, Chenglong Lu†, Linyan Yang, Guoqing Chen, Fu Zhang
[1]School of Computer Science and Engineering, Northeastern University, China
[2]Key Laboratory of Intelligent Computing of Medical Images,
Ministry of Education, Northeastern University, China.
{chenglonglu233,chenguoqing247}@gmail.com
{chengjingwei,zhangfu}@mail.neu.edu.cn
yanglinyanly@163.com

Abstract

Entity alignment (EA) aims to identify entities in different knowledge graphs (KGs) that represent the same real-world objects. Traditional EA methods typically embed entity information into vector space under the guidance of seed entity pairs, and align entities by calculating and comparing the similarity between entity embeddings. With the advent of large language models (LLMs), emerging methods are increasingly integrating LLMs with traditional methods to leverage external knowledge and improve EA accuracy. However, this integration also introduces additional computational complexity and operational overhead, and still requires seed pairs that are scarce and expensive to obtain. To address these challenges, we propose EasyEA, the first end-to-end EA framework based on LLMs that requires no training. EasyEA consists of three main stages: (1) Information Summarization, (2) Embedding and Feature Fusion, and (3) Candidate Selection. By automating the EA process, EasyEA significantly reduces the reliance on seed entity pairs while demonstrating superior performance across various datasets, covering cross-lingual, sparse, large-scale, and heterogeneous scenarios. Extensive experimental results show that EasyEA not only simplifies the EA process but also achieves state-of-the-art (SOTA) performance on diverse datasets, providing a promising solution for advancing EA tasks [1].

1 Introduction

Knowledge graphs (KGs) are structured knowledge bases widely used in tasks such as semantic search, recommendation systems, and question answering. These graphs typically represent real-world objects and their relation in the form of triples (entity-relation-entity or entity-attribute-value) (Sun et al., 2020). The goal of entity alignment (EA) is to identify equivalent entity pairs across different KGs that refer to the same real-world object (Fanourakis et al., 2023). As KGs differ in language, structure, and schema, EA has become a challenging task (Zhao et al., 2020; Fanourakis et al., 2023).

Traditional EA methods, such as translation-based methods, machine learning-based methods, and graph neural network (GNN)-based methods (Jiang et al., 2024a), rely on symbolic and structural features to align entities across KGs. These methods perform well in scenarios with consistent naming conventions or rich relation structures (Zhao et al., 2020). However, when applied to large or diverse KGs, they face significant challenges, particularly due to linguistic and structural heterogeneity. Furthermore, these methods require large amounts of labeled data for training and fail to incorporate external knowledge, both of which are crucial for accurate EA (Sun et al., 2020; Zhao et al., 2020; Fanourakis et al., 2023). Additionally, the black-box nature of embedding similarity calculations limits their interpretability and reduces adaptability to complex EA scenarios (Jiang et al., 2024a).

LLMs have significantly advanced various fields with their exceptional semantic understanding, contextual inference, and cross-lingual capabilities. These strengths make them particularly valuable for tackling challenges in EA, such as bridging the semantic gap between KGs and enriching limited entity knowledge. Recent EA methods combining LLMs with traditional methods have led to notable improvements in performance. Some methods focus on turning entity information into a common semantic form and using the search abilities of LLMs to align them efficiently, such as DERA (Wang and Chen, 2024) and Seg-Align (Yang et al., 2024a). Others leverage the reasoning power of LLMs to improve alignment accuracy and robustness through methods like multi-step reasoning and active learning, such as ChatEA (Jiang et al., 2024a) and LLMEA (Yang et al., 2024b).

*Corresponding author. †Equal contribution.
[1]Code: https://github.com/alusang/EasyEA-framework

Findings of the Association for Computational Linguistics: ACL 2025, pages 20981–20995
July 27 - August 1, 2025

While this combination enhances the accuracy of EA, it also introduces significant resource overhead. Specifically, methods that utilize LLMs require not only the computational resources necessary for smaller models but also the additional resources required by the LLM itself. This creates a challenge in balancing resource consumption with alignment accuracy (Jiang et al., 2024a). Furthermore, these methods rely on entity names to supplement or enrich entity information using LLMs, which can lead to potential data leakage (Wu et al., 2024), where sensitive or proprietary data associated with the entities may unintentionally influence the alignment process.

To cope with the complexity of the current EA task, we propose EasyEA, an efficient EA framework driven entirely by LLMs, aimed at overcoming the limitations of traditional models and hybrid models. EasyEA consists of three key stages: (1) Information Summarization. At this stage, we focus on using LLMs to extract semantic information from the KG data. The LLM summarizes the key attributes and relations of entities to capture their core semantic meanings. (2) Embedding and Feature Fusion. In this stage, we embed the summaries using LLMs and integrate the diverse feature embeddings obtained to construct a holistic and enriched representation of entities. (3) Candidate Selection. We propose a hierarchical strategy, which leverages multiple views of information, enabling the LLM to more accurately select the most appropriate target entities, thereby enhancing the accuracy and reliability of EA.

Through extensive experiments on multiple datasets, EasyEA demonstrates excellent performance, surpassing existing state-of-the-art (SOTA) models. Unlike traditional methods, EasyEA eliminates the need for manual seed entity pair construction and additional model training, significantly improving efficiency while ensuring high-quality EA results. The main contributions of our framework are:

- We introduce the first fully LLM-based EA framework EasyEA, eliminating the reliance on traditional methods and enabling an end-to-end EA process driven entirely by LLMs.
- By relying solely on LLMs, EasyEA removes the need for seed entity pair construction and eliminates the need for additional training, significantly reducing the manual effort required in traditional EA methods.
- EasyEA framework achieves SOTA performance on widely-used datasets, including DBP15K, ICEWS, SRPRS, and DWY, demonstrating its effectiveness and robustness in challenging scenarios such as cross-lingual alignment, large-scale KGs, heterogeneous KGs, and sparse datasets.

2 Related Works

EA methods can generally be classified into four categories: translation-based methods, machine learning-based methods, GNN-based methods, and LLM-enhanced methods.

Translation-Based Methods. Translation-based methods, such as MTransE (Chen et al., 2017), BootEA (Sun et al., 2018), and Transedge (Sun et al., 2019), represent entities and relations in a low-dimensional vector space. In these models, a relation in KGs is treated as a translation mapping the head entity vector to the tail entity vector (Zhang et al., 2022). These methods align entities by minimizing the distance between the vectors of aligned entities. While effective in homogeneous KGs, these methods face challenges in more complex or heterogeneous graph structures, where relations can be more complicated (Zhang et al., 2022). Furthermore, translation-based models often struggle with cross-lingual or sparse data settings, where the embeddings may fail to fully capture the diversity and complexity of the data.

Machine Learning-Based Methods. Machine learning-based methods introduce supervised or semi-supervised learning techniques, using seed entity pairs from KGs to train classifiers or regression models. Notable machine learning-based methods include BERT-INT (Tang et al., 2020), and Simple-HHEA (Jiang et al., 2024b), which leverage different machine learning techniques to enhance EA performance. However, these methods are heavily dependent on the quality and quantity of seed entity pairs, leading to high labeling costs. Moreover, their performance can be constrained in cross-lingual or sparse data scenarios, where labeled data is often scarce (Fanourakis et al., 2023).

Graph neural network (GNN)-Based Methods. GNN-based methods, such as GCN-Align (Wang et al., 2018), MuGNN (Cao et al., 2019) and RDGCN (Wu et al., 2019), explicitly model the graph structure of KGs, learning high-order features of nodes and their neighbors. These methods show certain advantages in capturing both local

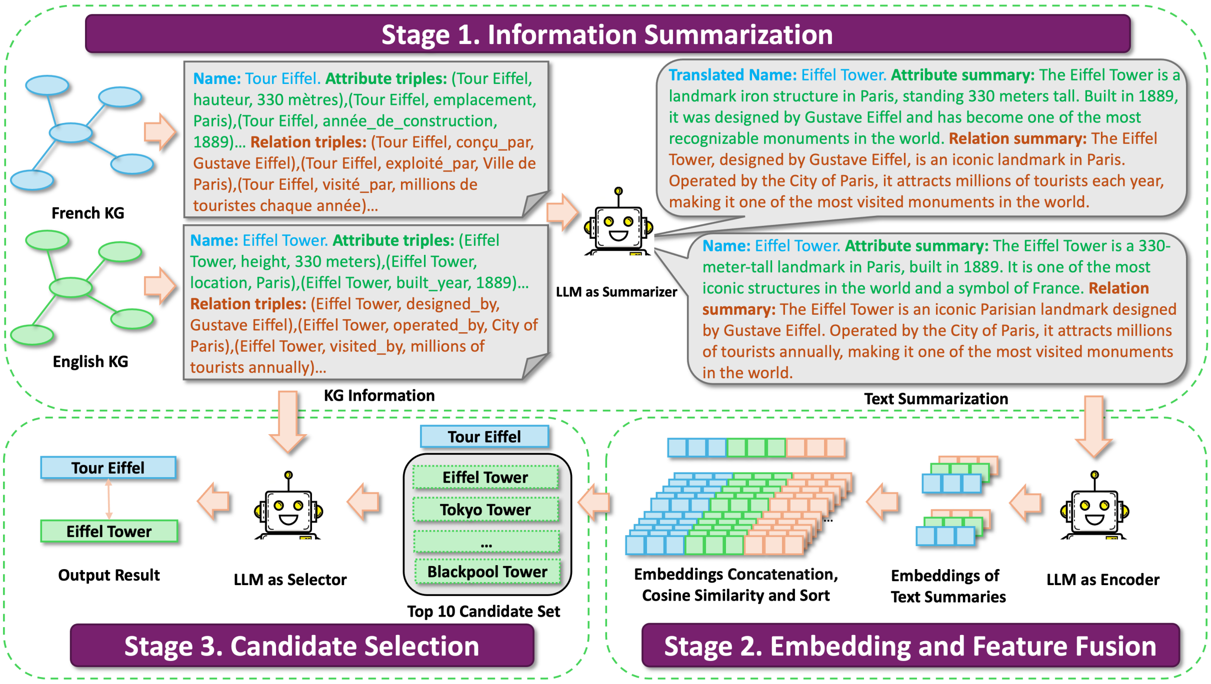

Figure 1: The framework of EasyEA we proposed is mainly divided into three stages: (1) Information Summarization; (2) Embedding and Feature fusion; (3) Candidate selection.

and global structural information, making them effective for EA in complex graph environments. However, their dependence on labeled data and high computational complexity limits their scalability, especially in large-scale or heterogeneous datasets.

LLM-Enhanced Methods. With the advent of LLMs, EA methods have evolved into hybrid frameworks that combine the strengths of traditional models with the semantic capabilities of LLMs. ChatEA (Jiang et al., 2024a) enhances candidate selection through iterative reasoning, while Seg-Align (Yang et al., 2024a) integrates small language models for feature extraction and LLMs for cross-lingual alignment. LLMEA (Yang et al., 2024b) combines LLM insights with structural embeddings to improve consistency in alignment. Additionally, DERA (Wang and Chen, 2024) encodes entity information into text representations, improving retrieval and reducing structural-semantic inconsistencies, and LLM4EA (Chen et al., 2024) integrates LLM-encoded knowledge with traditional embeddings to enhance entity quality. These approaches highlight the potential of LLMs but also introduce challenges, such as the need for computational resources (Jiang et al., 2024a). Additionally, these methods rely on LLMs to enhance or supplement entity information, which could lead to unintended data leakage (Wu et al., 2024).

To address the limitations of traditional methods and hybrid methods, we propose EasyEA, a fully LLM-based EA framework. By removing reliance on traditional techniques, EasyEA significantly reduces complexity while achieving competitive performance across a variety of challenging datasets.

3 Problem Definition

A **Knowledge Graph (KG)** is represented as $KG = (V, R, A, \mathbb{V}, T)$, where V, R, A, $\mathbb{V}$, and T represent entities, relations, attribute types, attribute values, and triples, respectively. Each entity $v \in V$ represents a real-world object or concept, and each relation $r \in R$ represents a relation between two entities. The set of attribute types is denoted as A, and the set of attribute values is denoted as $\mathbb{V}$. The set of triples T can be further divided into two categories: relation triples and attribute triples. Relation triples are represented as $T^R = \{t^r = (v_i, r_{ij}, v_j) \mid v_i, v_j \in V, r_{ij} \in R\}$, where r_{ij} represents a specific relation between entities v_i and v_j. Attribute triples are represented as $T^A = \{t^a = (v_i, a_k, a_v) \mid v_i \in V, a_k \in A, a_v \in \mathbb{V}\}$, where $a_k \in A$ represents the attribute type (e.g., "name", "age"), and $a_v \in \mathbb{V}$ represents the corresponding attribute value. Consequently, the set of triples T in KG can be expressed as the union of relation and attribute triples, i.e., $T = T^R \cup T^A$.

Entity Alignment (EA) involves identifying equivalent entities across different KGs. Given two KGs, $KG_1 = (V_1, R_1, A_1, \mathbb{V}_1, T_1)$ and $KG_2 = (V_2, R_2, A_2, \mathbb{V}_2, T_2)$, the task is to find a set of aligned entity pairs $EA(KG_1, KG_2) = \{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2, v_1 \approx v_2\}$, where $\approx$ denotes semantic equivalence. In EA, entities v_1 and v_2 are considered aligned when they represent the same real-world concept or object, despite potentially different identifiers, attributes, or structures in the respective KGs.

4 Method

In this section, we describe the core process of the EasyEA framework, which is divided into three main stages: (1) Information Summarization, (2) Embedding and Feature Fusion, and (3) Candidate Selection. The framework diagram, shown in Figure 1, illustrates the overall process of these stages.

Stage 1. Information Summarization

In KGs, each entity is associated with various types of information, such as its name, relations, attributes, and temporal data. While datasets may vary in the types of entity information they contain, LLMs excel at extracting semantic representations and summarizing them concisely. In the EasyEA framework, we focus on three key types of entity information: entity name, attributes, and relations.

Initially, entity names, attribute triples, and relation triples are extracted from the KGs. The entity names are translated into English using LLMs, while attribute and relation triples are consolidated into separate texts to represent entity attributes and relations. The LLMs then summarize these texts, compressing the information into no more than 100 words.

Our framework uses only entity name translation to avoid potential data leakage (Wu et al., 2024), focusing entirely on the entities themselves without involving any information mining or background inference. This strategy effectively prevents the leakage of sensitive information and reduces the risk of generating content related to the entity background. Additionally, the method leverages a key advantage of LLMs—summarization—to efficiently extract information from KGs, concentrating on existing, verifiable data rather than generating new content. By limiting the summaries to no more than 100 words, we ensure that the output of the model emphasizes more distinctive features, minimizing hallucinations (Sriramanan et al., 2024) and further enhancing the differentiation of entity characteristics.

Stage 2. Embedding and Feature Fusion

This Stage aims to enhance EA performance by integrating multiple views of information to create a more comprehensive entity representation. EA datasets, such as DBP15K and ICEWS, exhibit distinct characteristics. For instance, in the ZH-EN subset of DBP15K, strong performance can be achieved using only attribute information, while the ICEWS-WIKI dataset performs well with name information alone (Jiang et al., 2024b). These observations highlight the need to combine diverse information sources for a more complete entity representation.

To address this, we first encode the translated entity names, attribute summaries, and relation summaries in Stage 1 into embeddings E^N, E^A, and E^R. Once the embeddings are generated, we propose a feature fusion strategy where these embeddings are concatenated to form the holistic entity embedding E, as shown in equation 1.

$$E = E^N \parallel E^A \parallel E^R \tag{1}$$

This approach effectively leverages the complementary strengths of each feature type, ensuring a more comprehensive and accurate entity representation.

Stage 3. Candidate Selection

In this stage, we first compute the cosine similarity between entity embedding vectors from Stage 2. Based on these ranked similarities, the top 10 most similar candidate entities are selected to form a candidate set. This refined set is processed by LLMs to select the most likely target entity, with the final selection corresponding to Hits@1. For each candidate, its name, along with three randomly selected attribute triples and three randomly selected relation triples from the KGs, are provided as input to the LLM.

The entity selection follows a hierarchical strategy we propose: the LLM first uses name information to identify the target. If name data is insufficient, attribute triples are used to refine the selection. If further refinement is needed, relation triples are used as a final step. The LLM autonomously determines the "insufficiency" at each stage based on the completeness and relevance of the available data, without relying on predefined thresholds.

This strategy prioritizes the most informative features. By focusing on name information first, we maximize its potential for accurate entity identification. When name information is insufficient, attribute and relation triples offer additional context, improving the accuracy of entity selection.

The decision to select 10 candidate entities is based on two factors: first, Hits@10 is a standard metric in evaluation, ensuring consistency with common practices; second, the reasoning capability of LLMs declines with input size, and too many candidates can reduce accuracy (Wang et al., 2024). The algorithmic flow of EasyEA is outlined in Appendix A.4, with specific prompts provided in Appendix A.7.

5 Experiments

5.1 Research Questions

RQ1: Can LLMs effectively act as summarizers to enhance the alignment process in EA?

RQ1 explores whether LLMs can serve as summarizers to enhance the EA process by refining entity information. We evaluate their ability to summarize key entity attributes and relations, improving the overall alignment across diverse datasets.

RQ2: Can LLMs effectively serve as a good encoder for generating high-quality entity embeddings in EA?

This question investigates whether LLMs can be used as encoders to generate high-quality entity embeddings for EA, comparing their performance with traditional methods. We focus on the quality, consistency, and generalization of embeddings generated by LLMs.

RQ3: How can LLMs function as selectors to improve candidate entity selection during the EA process?

RQ3 investigates how LLMs can function as selectors to enhance the selection of the most relevant candidate entities in the EA process. We explore how LLMs, through techniques like hierarchical filtering or ranking, can improve the precision and efficiency of candidate selection.

5.2 Experimental Setup

5.2.1 Datasets

DBP15K (ZH-EN, JA-EN, FR-EN) (Tang et al., 2020) is a widely used cross-lingual dataset for testing EA across KGs in different languages, focusing on overcoming linguistic barriers. **SRPRS** (EN-DE, EN-FR, DBP-WIKI15K, DBP-YAGO15K) (Zeng et al., 2020) consists of datasets designed to evaluate EA in sparse, heterogeneous graph structures, addressing challenges in low-resource settings. **ICEWS** (ICEWS-WIKI, ICEWS-YAGO) (Jiang et al., 2024b) includes datasets characterized by high heterogeneity in graph structures and information density, testing the adaptability of the framework to heterogeneous KG. **DWY** (DBP-WIKI100K, DBP-YAGO100K) (Liu et al., 2022) presents the main challenge of large scale, which imposes significant computational demands for processing and alignment, requiring substantial memory and processing power. More details about these datasets are shown in Appendix A.1.

5.2.2 Baselines

To comprehensively evaluate the performance of the proposed EasyEA method, we compare it with a diverse set of existing EA methods. These baselines include both well-established techniques and recent innovations, reflecting a broad spectrum of methods in the field. The selected baselines are grouped into four categories: (1) **Translation-Based Methods**: MTransE (Chen et al., 2017), BootEA (Sun et al., 2018), TransEdge (Sun et al., 2019). (2) **GNN-Based Methods**: GCN-Align (Wang et al., 2018), RDGCN (Wu et al., 2019), MuGNN (Cao et al., 2019), KECG (Li et al., 2019), Dual-AMN (Mao et al., 2021), CEA (Zeng et al., 2020), EPEA (Wang et al., 2020), Selfkg (Liu et al., 2022). (3) **Machine Learning-Based Methods**: BERT-INT (Tang et al., 2020), MRAEA (Mao et al., 2020), MultiKE (Zhang et al., 2019), FuAlign (Wang et al., 2023), JAPE (Sun et al., 2017), NAEA (Zhu et al., 2019), RSN4EA (Guo et al., 2019), Simple-HHEA (Jiang et al., 2024b). (4) **LLM-Enhanced methods**: LLM4EA (Chen et al., 2024), DERA (Wang and Chen, 2024), LLMEA (Yang et al., 2024b), ChatEA (Jiang et al., 2024a), Seg-Align (Yang et al., 2024a). These baselines span a wide range of methodologies, from traditional methods to LLM-enhanced methods, providing a robust basis for evaluating EasyEA's performance against SOTA methods.

5.3 Main Experimental Results

The experimental results of EasyEA on the DBP15K, ICEWS, SRPRS, and DWY datasets are summarized in Tables 1, 2, 3, and 4. On the DBP15K dataset, EasyEA achieves Hits@1 scores of **0.997**, **0.995**, and **0.998** for ZH-EN, JA-EN, and FR-EN, respectively, with perfect Hits@10 (**1.000**). On the ICEWS dataset, EasyEA achieves Hits@1

Models	DBP15K$_{ZH-EN}$			DBP15K$_{JA-EN}$			DBP15K$_{FR-EN}$		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	0.308	0.614	0.364	0.279	0.575	0.349	0.247	0.577	0.360
GCN-Align	0.413	0.744	0.549	0.399	0.745	0.546	0.411	0.772	0.530
BootEA	0.629	0.848	0.703	0.622	0.854	0.701	0.653	0.874	0.731
RDGCN	0.708	0.846	0.746	0.767	0.895	0.812	0.873	0.950	0.901
Dual-AMN	0.861	0.964	0.901	0.892	0.978	0.925	0.954	0.994	0.970
LLMEA	0.898	0.923	-	0.911	0.946	-	0.957	0.977	-
Seg-Align	0.953	-	-	0.907	-	-	0.987	-	-
BERT-INT	0.968	0.990	0.977	0.964	0.991	0.975	0.990	0.997	0.993
ChatEA	-	-	-	-	-	-	0.990	1.000	0.995
DERA	0.985	0.997	0.990	0.994	0.999	0.996	0.996	0.999	0.997
EasyEA	**0.997**	**1.000**	**0.996**	**0.995**	**1.000**	**0.997**	**0.998**	**1.000**	**0.999**

Table 1: Main experimental results of EasyEA on DBP15k datasets.

Models	ICEWS-WIKI			ICEWS-YAGO		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	0.021	0.158	0.068	0.012	0.084	0.040
GCN-Align	0.046	0.184	0.093	0.017	0.085	0.038
RDGCN	0.064	0.202	0.096	0.029	0.097	0.042
BootEA	0.072	0.275	0.139	0.020	0.120	0.056
Dual-AMN	0.083	0.281	0.145	0.031	0.144	0.068
FuAlign	0.257	0.570	0.361	0.326	0.604	0.423
BERT-INT	0.561	0.700	0.607	0.756	0.859	0.793
Simple-HHEA	0.720	0.872	0.754	0.847	0.915	0.870
ChatEA	0.880	0.945	0.912	0.935	0.955	0.944
EasyEA	**0.995**	**0.999**	**0.996**	**0.994**	**0.998**	**0.996**

Table 2: Main experimental results of EasyEA on ICEWS datasets.

of **0.995** (WIKI) and **0.994** (YAGO), outperforming models like ChatEA and Simple-HHEA. The SRPRS results show Hits@1 of **0.998** (EN-DE), **0.996** (EN-FR), **1.000** (DBP-YAGO), and **1.000** (DBP-WIKI). Similarly, EasyEA achieves perfect scores across all metrics on the DWY datasets, with Hits@1, Hits@10, and MRR of **1.000** on both WIKI and YAGO, outperforming all other models.

These results highlight the strong performance of EasyEA, confirming that LLM-enhanced methods can serve as a superior alternative to traditional models and hybrid models for EA tasks. This suggests that LLMs, with their ability to process unstructured data and provide richer semantic understanding, outperform conventional models. EasyEA demonstrates excellent adaptability across different languages and structures, showcasing its effectiveness in various scenarios.

The main experimental results are obtained using GPT-3.5-Turbo for summarization, Llama3-8B-Instruct for embedding, and GPT-4-Turbo for further optimization.

5.4 Ablation Experiment

5.4.1 Ablation Experiments of Summarization

To evaluate the effectiveness and generalizability of our summarization strategy, we conducted two ablation experiments. First, we replaced GPT-3.5-Turbo with Llama3-8B-Instruct in the summarization module to compare the performance of different LLMs. Second, we removed the summarization module entirely (*w/o summarization*) to assess its overall contribution to entity alignment.

As shown in Table 5, replacing GPT-3.5-Turbo with Llama3-8B-Instruct led to a slight decrease in performance. Specifically, Llama3-8B-Instruct achieved a Hits@1 of 0.991, Hits@10 of 1.000, and an MRR of 0.991, compared to EasyEA's Hits@1 of **0.997** and MRR of **0.996**. In contrast, completely removing the summarization component resulted in a substantial performance drop, with Hits@1 falling to 0.921 and MRR to 0.927.

These results demonstrate that our LLM-guided summarization strategy significantly contributes to alignment accuracy by transforming redundant at-

Models	SRPRS$_{EN-DE}$			SRPRS$_{EN-FR}$			SRPRS$_{DBP-YAGO}$			SRPRS$_{DBP-WIKI}$		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	0.107	0.248	0.160	0.213	0.447	0.290	0.196	0.401	0.270	0.188	0.382	0.260
MuGNN	0.245	0.431	0.310	0.131	0.342	0.208	0.175	0.381	0.240	0.151	0.366	0.220
NAEA	0.307	0.535	0.390	0.177	0.416	0.260	0.195	0.451	0.280	0.182	0.429	0.260
GCN-Align	0.385	0.600	0.460	0.243	0.522	0.340	0.319	0.586	0.410	0.291	0.556	0.380
KECG	0.444	0.707	0.540	0.298	0.616	0.403	0.350	0.651	0.450	0.323	0.646	0.430
RSN4EA	0.484	0.729	0.570	0.350	0.636	0.440	0.393	0.665	0.490	0.391	0.663	0.480
BootEA	0.503	0.732	0.580	0.365	0.649	0.460	0.381	0.651	0.470	0.384	0.667	0.480
TransEdge	0.556	0.753	0.630	0.400	0.675	0.490	0.443	0.699	0.530	0.461	0.738	0.560
MRAEA	0.594	0.818	0.666	0.460	0.768	0.559	0.485	0.768	0.574	0.509	0.795	0.597
RDGCN	0.779	0.886	0.820	0.672	0.767	0.710	0.990	0.997	0.990	0.974	0.994	0.980
Dual-AMN	0.891	0.972	0.923	0.802	0.932	0.851	0.518	0.795	0.613	0.546	0.813	0.635
BERT-INT	0.986	0.988	0.990	0.971	0.975	0.970	1.000	1.000	1.000	0.996	0.997	1.000
EasyEA	**0.998**	**1.000**	**0.999**	**0.996**	**0.998**	**0.992**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table 3: Main experimental results of EasyEA on SRPRS datasets.

Models	DWY$_{DBP-WIKI}$			DWY$_{DBP-YAGO}$		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	0.281	0.520	0.363	0.252	0.493	0.334
JAPE	0.318	0.589	0.411	0.236	0.484	0.320
GCN-Align	0.506	0.772	0.600	0.597	0.838	0.682
MuGNN	0.616	0.897	0.714	0.741	0.937	0.810
RDGCN	0.623	0.805	0.684	0.936	0.973	0.950
BootEA	0.748	0.898	0.801	0.761	0.894	0.808
NAEA	0.767	0.917	0.817	0.778	0.912	0.821
Dual-AMN	0.869	0.969	0.908	0.907	0.981	0.935
LLM4EA	0.898	0.979	0.929	0.979	0.996	0.985
MultiKE	0.914	0.951	0.928	0.880	0.953	0.906
EPEA	0.975	0.981	0.977	1.000	1.000	1.000
SelfKG	0.983	0.998	-	1.000	1.000	-
ChatEA	0.995	1.000	0.998	-	-	-
EasyEA	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table 4: Main experimental results of EasyEA on DWY datasets.

tribute information into concise and semantically meaningful representations. Even though Llama3-8B-Instruct slightly underperforms GPT-3.5-Turbo, it still delivers strong results, indicating that our approach is robust across different LLMs. In contrast, removing the summarization step weakens the ability to identify distinguishing features of entities, leading to a noticeable drop in performance, with the accuracy decreasing from **0.997** to 0.921. Overall, these findings validate the effectiveness and adaptability of our summarization method (RQ1).

Settings	DBP15K$_{ZH-EN}$		
	Hits@1	Hits@10	MRR
Llama3-8B-Instruct	0.991	1.000	0.991
w/o summarization	0.921	0.932	0.927
EasyEA	**0.997**	**1.000**	**0.996**

Table 5: Ablation results under different summarization settings

5.4.2 Ablation Experiments of Features Fusion

To evaluate the contribution of different types of entity information to embedding quality, we conducted an ablation experiment where one type of information was excluded while retaining the other two. The results, in Table 6, show that the fusion of all three types yields the best performance, with Hits@1 of **0.997**, Hits@10 of **1.000**, and MRR of **0.996**.

Removing name information (*w/o name*) caused a slight decrease, with Hits@1 dropping to 0.994. The absence of relation information (*w/o relation*) led to a similar performance drop, with Hits@1 dropping to 0.990. However, removing attribute information (*w/o attribute*) resulted in the most significant performance degradation, with Hits@1 falling to 0.977.

This result strongly demonstrate the superiority of feature fusion strategy of EasyEA and emphasize the importance of combining multiple types of information for optimal EA.

Settings	DBP15K$_{ZH\text{-}EN}$		
	Hits@1	Hits@10	MRR
EasyEA	**0.997**	**1.000**	**0.996**
w/o name	0.994	1.000	0.995
w/o attribute	0.977	0.989	0.963
w/o relation	0.990	0.999	0.991

Table 6: Results of using different information for embedding

5.4.3 Comparative Experiments of Embedding with Different LLMs

We evaluated EasyEA's ability to generalize in the embedding stage by testing it with different LLMs.

As presented in Table 7, EasyEA demonstrates exceptional performance even with medium-sized LLMs (7B–8B parameters). Notably, with LLama3-8B-Instruct, EasyEA achieves SOTA results, with Hits@1 reaching **0.997**, Hits@10 achieving a perfect **1.000**, and MRR scoring **0.996**. Llama2-7B-Chat and Mistral-7B-Instruct also deliver strong results, with Hits@1 and Hits@10 surpassing 0.99.

The results demonstrate that LLMs are effective encoders for EA, ensuring strong performance across a range of models (RQ2). This highlights EasyEA's adaptability and potential for real-world applications, where it maintains robust performance even when using smaller LLMs in resource-constrained settings.

Settings	DBP15K$_{ZH\text{-}EN}$		
	Hits@1	Hits@10	MRR
EasyEA	**0.997**	**1.000**	**0.996**
Llama2-7B-Chat	0.992	0.998	0.991
Mistral-7B-Instruct	0.991	0.997	0.991

Table 7: Results of using various LLMs for embedding

5.4.4 Ablation Experiments for Candidate Selection

We conducted ablation experiments to assess the impact of using LLMs to select the best matching entities in Stage 3 of EasyEA.

The experimental results in Table 8 show a consistent improvement in the Hits@1 scores when LLMs are used as a selector. For example, with GPT-3.5-Turbo + LLama3-8B, the Hits@1 score improves from 0.994 to **0.997**. Similarly, the Hits@1 score for LLama3-8B + LLama3-8B increases from 0.986 to **0.991**, for Llama2-7B + Llama2-7B from 0.948 to **0.983**, and for Mistral-7B + Mistral-7B from 0.931 to **0.981**.

These results demonstrate that LLM-based reasoning significantly improves EA performance, particularly when initial Hits@1 scores are lower. However, when the initial score is already high, the performance gain is less pronounced. This is due to our focus on a simplified setup that avoids complex Prompt Engineering, aiming to validate the method's feasibility. Overall, the findings highlight the effectiveness of using LLMs as selectors in the EA process(RQ3).

Settings	DBP15K$_{ZH\text{-}EN}$	
	Hits@1 w/ llm	Hits@1 w/o llm
EasyEA	**0.997**	0.994
LLama3-8B + LLama3-8B	**0.991**	0.986
Llama2-7B + Llama2-7B	**0.983**	0.948
Mistral-7B + Mistral-7B	**0.981**	0.931

Table 8: Results of whether to use LLMs reasoning

5.4.5 Ablation Experiments of Single Feature Retention

we conducted ablation by removing two types of information and retaining only one type for evaluation. As shown in Table 9, when only name information (*w/ name*) is retained, the performance dropped significantly, with Hits@1 falling to 0.842. Similarly, when only relation information (*w/ relation*) is used, the performance is also significantly lower, with Hits@1 dropping to 0.973. In contrast, retaining only attribute information (*w/ attribute*) resulted in relatively higher performance, with Hits@1 of 0.991, close to the performance of the full model.

These results strongly demonstrate the superiority of EasyEA's feature fusion strategy and emphasize the importance of combining multiple types of information for optimal EA.

Settings	DBP15K$_{ZH\text{-}EN}$		
	Hits@1	Hits@10	MRR
EasyEA	**0.997**	**1.000**	**0.996**
w/ name	0.842	0.879	0.832
w/ attribute	0.991	0.998	0.992
w/ relation	0.973	0.990	0.956

Table 9: Results of using one information for embedding

5.5 Efficiency Analysis

Compared to traditional methods, the EasyEA framework significantly simplifies the EA process and improves efficiency. Traditional methods often require constructing seed entity pairs, which involves considerable manual effort and complex model training. Moreover, the variety of models and complex code structures increase learning costs. In contrast, EasyEA leverages the widespread use of LLMs and can be implemented with simple, easy-to-understand code. There is no need to construct seed entity pairs or perform model training. By simply extracting dataset information and passing it to the LLM, EasyEA delivers excellent alignment results.

5.6 Cost Analysis

In this section, we analyze the costs of EasyEA on DBP15K$_{\text{ZH-EN}}$ dataset. EasyEA utilizes LLMs across three stages: Information Summarization, Embedding and Feature Fusion, and Candidate Selection. The detailed cost breakdown is as follows:

Stage 1: Information Summarization. In this stage, we use GPT-3.5 Turbo to summarize the entity information. For the DBP15K$_{\text{ZH-EN}}$ dataset, with 30k entities and each containing an average of 20 triples (10 tokens per triple), the total input data is approximately 12M tokens. The output for each summary does not exceed 100 tokens, leading to an output of 6M tokens. The total cost for this stage is approximately $3.6, considering the API fee of $0.2 per million tokens.

Stage 2: Embedding and Feature Fusion. This stage involves local inference using a deployed LLM for embedding. Running on an A100-40G GPU, this process takes less than 30 minutes, costing about $0.3.

Stage 3: Candidate Selection. For candidate selection, the LLM API is used again, where up to 1.5k unmatched entities are processed. The total token usage for this stage is about 1M tokens, leading to a cost of $0.2.

Total Cost of EasyEA. Thus, the total cost of processing the DBP15K$_{\text{ZH-EN}}$ dataset using EasyEA is approximately $4.1.

6 Conclusion

This work primarily explores the feasibility of using LLMs for EA without relying on traditional models. We propose the EasyEA framework, which relies solely on LLMs for EA, and validate its feasibility through extensive experiments and ablation analysis, achieving excellent alignment results. This method eliminates the training requirements of traditional models and the need for seed entity pair construction, making EA simpler and more efficient. Additionally, we evaluate EasyEA's performance on multiple common datasets, achieving strong results, and introduce a simple and efficient candidate selection method to further enhance EA efficiency.

Limitations

Although EasyEA is simple, efficient, and achieves excellent EA results, it has some limitations. For example: (1) Limitations of structural information in text embedding. As LLMs are generative models, they struggle to accurately understand and utilize the structural information of entities, leading to an incomplete exploration of this aspect. There is significant research potential here; (2) Hardware resource requirements. While LLM-based methods are faster and more efficient than traditional models, they still require certain hardware resources. We believe this limitation will gradually be overcome with ongoing advancements in hardware and LLMs; (3) When Hits@1 is already very high, further refinement with LLMs provides minimal improvements. This indicates that in such cases, LLMs have limited impact. Exploring how LLMs can still offer significant gains despite high initial performance is an area worth further research.

Ethics Statement

To the best of our knowledge, this work does not involve any discrimination, social bias, or private data. All the datasets are constructed from open-source KGs such as Wikidata, YAGO, ICEWS, and DBpedia. Therefore, we believe that our work complies with the ACL Ethics Policy.

Acknowledgments

The authors sincerely thank the reviewers for their valuable comments, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057).

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *CoLM*.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1452–1461. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1511–1517. ijcai.org.

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. 2024. Entity alignment with noisy annotations from large language models. *arXiv preprint arXiv:2405.16806*.

Nikolaos Fanourakis, Vasilis Efthymiou, Dimitris Kotzinos, and Vassilis Christophides. 2023. Knowledge graph embedding methods for entity alignment: experimental review. *Data Mining and Knowledge Discovery*, 37(5):2070–2137.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *International conference on machine learning*, pages 2505–2514. PMLR.

Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. 2024a. Unlocking the power of large language models for entity alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7566–7583. Association for Computational Linguistics.

Xuhui Jiang, Chengjin Xu, Yinghan Shen, Yuanzhuo Wang, Fenglong Su, Zhichao Shi, Fei Sun, Zixuan Li, Jian Guo, and Huawei Shen. 2024b. Toward practical entity alignment method design: Insights from new highly heterogeneous knowledge graph datasets. In *Proceedings of the ACM on Web Conference 2024*, pages 2325–2336.

Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. Association for Computational Linguistics.

Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2022. Selfkg: Self-supervised entity alignment in knowledge graphs. In *Proceedings of the ACM Web Conference 2022*, pages 860–870.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In *Proceedings of the Web Conference 2021*, pages 821–832.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 420–428. ACM.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. Llm-check: Investigating detection of hallucinations in large language models. *Advances in Neural Information Processing Systems*, 37:34188–34216.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. 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*, volume 10587 of *Lecture Notes in Computer Science*, pages 628–644. Springer.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, 2018.

Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 612–629. Springer.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(11):2326–2340.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. BERT-INT: A bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3174–3180. ijcai.org.

Chenxu Wang, Zhenhao Huang, Yue Wan, Junyu Wei, Junzhou Zhao, and Pinghui Wang. 2023. Fualign: Cross-lingual entity alignment via multi-view representation learning of fused knowledge graphs. *Information Fusion*, 89:41–52.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi.

2024. Beyond the limits: A survey of techniques to extend the context length in large language models. *arXiv preprint arXiv:2402.02244*.

Zhichun Wang and Xuan Chen. 2024. Dera: Dense entity retrieval for entity alignment in knowledge graphs. *arXiv preprint arXiv:2408.01154*.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 349–357.

Zhichun Wang, Jinjian Yang, and Xiaoju Ye. 2020. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1672–1680.

Yonghao Wu, Zheng Li, Jie M Zhang, and Yong Liu. 2024. Condefects: A complementary dataset to address the data leakage concern for llm-based fault localization and program repair. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, pages 642–646.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5278–5284. ijcai.org.

Linyan Yang, Jingwei Cheng, and Fu Zhang. 2024a. 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*, pages 8122–8138.

Linyao Yang, Hongyang Chen, Xiao Wang, Jing Yang, Fei-Yue Wang, and Han Liu. 2024b. Two heads are better than one: Integrating knowledge from knowledge graphs and large language models for entity alignment. *arXiv preprint arXiv:2401.16960*.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2020. Collective entity alignment via adaptive features. In *2020 IEEE 36th international conference on data engineering (ICDE)*, pages 1870–1873. IEEE.

Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5429–5435. ijcai.org.

Rui Zhang, Bayu Distiawan Trisedya, Miao Li, Yong Jiang, and Jianzhong Qi. 2022. A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. *The VLDB Journal*, 31(5):1143–1168.

Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian M Suchanek. 2020. An experimental study of state-of-the-art entity alignment approaches. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2610–2625.

Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *ijcai*, pages 1943–1949.

A Appendix

A.1 Statistical Data of DBP15K, SRPRS, ICEWS and DWY

All datasets are selected due to their broad range of EA challenges, including cross-lingual, sparsity, heterogeneity, and large scale. Together, they provide a comprehensive benchmark to assess the effectiveness of EasyEA.

The information of DBP15K, SRPRS and DWY are shown in Table 13. The DWY dataset utilized in this work is divided into two major subsets: DBP-WIKI and DBP-YAGO, with each subset containing 100,000 pairs of aligned entities. In the DBP-WIKI subset, entities from the Wikidata portion are identified by indices (e.g., Q123) instead of URLs containing entity names. To obtain the actual entity names, we use the Wikidata API for Python (Liu et al., 2022).

The information of ICEWS is shown in Table 14, following is an introduction to the dataset.

Facts represents the total number of facts in the dataset. Facts are the basic units of a knowledge graph, expressed as triples comprising a head entity, a relation, and a tail entity.

Density measures the concentration of edges (relations) in the graph. It reflects the complexity and connectivity of the knowledge graph, with higher values indicating denser structures.

Anchors specifies the number of anchor links, which are aligned entity pairs. These are crucial for training and evaluating EA models.

Overlapping Ratio describes the proportion of alignable entities between the two graphs. A lower overlapping ratio signifies higher heterogeneity and greater alignment challenges.

Structure Similarity quantifies the similarity of the neighborhood structures of aligned entities across the graphs. Lower values indicate more significant structural differences.

Temporal indicates whether the dataset includes temporal information, capturing timestamps for facts and enabling temporal-aware EA research.

A.2 Model Selection and Parameters

In this experiment, we selected the Llama, Mistral, and GPT series as backbone models. These models are open-source or widely adopted and have demonstrated remarkable performance in related fields. Specifically, the Llama and Mistral models are employed in the embedding stage, as prior studies have shown their effectiveness and suitability for such tasks (BehnamGhader et al., 2024). These models have been extensively used in the literature, with their performance validated through numerous experiments. The GPT family is employed for summarization, reasoning, and selecting target entities, primarily due to its autoregressive architecture, which excels at handling complex dependencies and generating coherent, contextually relevant predictions. Additionally, the GPT models leverage their extensive knowledge base, acquired through large-scale pre-training, enabling them to achieve high accuracy in summarization, reasoning, and entity selection tasks. See Table 10 for details.

Usage	Models
Summarization	GPT-3.5-Turbo, Llama3-8B-Instruct, Llama2-7B-Chat, Mistral-7B-Instruct
Embedding	Llama3-8B-Instruct, Llama2-7B-Chat, Mistral-7B-Instruct
Reasoning	GPT-3.5-turbo, GPT-4o, GPT-4-trubo, Llama3-70B

Table 10: Model selection of EasyEA

For the experimental setup, we adhered strictly to the hyperparameter configurations recommended in the original publications for the baseline models, with only minor adjustments made to parameters such as max_tokens $= 4096$ and temperature $= 0.3$. All experiments are conducted in the PyTorch development environment, using an Ubuntu machine equipped with an 40GB NVIDIA A100 GPU. This hardware and software configuration ensured both the efficiency and stability of the experiments.

A.3 Evaluation Metrics

We use Hits@K and MRR as evaluation metrics because they are the most classic and commonly used in EA. Hits@K measures the proportion of correct entities within the top K predicted results, reflecting the model's ranking accuracy. MRR evaluates the average of the reciprocals of the ranks of the first correct entity, reflecting the model's ability to prioritize relevant entities. Together, these metrics provide a comprehensive assessment of model performance in EA tasks.

Algorithm 1 EasyEA Algorithm

Input: Entity names: n_1, n_2, attribute triples: T_1^A, T_2^A of entities v_1 and v_2, relation triples: T_1^R, T_2^R of entities v_1 and v_2
Output: The ID of the most likely target entity v_2 for each source entity v_1
Stage 1: Translation and Summarization
$N_1, N_2 \leftarrow \text{Translate}(n_1, n_2)$
$S_1^A, S_2^A \leftarrow \text{Summarize}(T_1^A, T_2^A)$
$S_1^R, S_2^R \leftarrow \text{Summarize}(T_1^R, T_2^R)$
Stage 2: Embedding and Fusion
$E_1^N, E_2^N \leftarrow \text{EmbedNames}(N_1, N_2)$
$E_1^A, E_2^A \leftarrow \text{EmbedAttributes}(S_1^A, S_2^A)$
$E_1^R, E_2^R \leftarrow \text{EmbedRelations}(S_1^R, S_2^R)$
$E_1 \leftarrow \text{Concat}(E_1^N, E_1^A, E_1^R)$
$E_2 \leftarrow \text{Concat}(E_2^N, E_2^A, E_2^R)$
Stage 3: Candidate Selection
$Cand \leftarrow$ Top-10 by Cosine Similarity(E_1, E_2)
$I \leftarrow$ Concat(id, name, $3 * t^a, 3 * t^r$)
$v_2 \leftarrow$ Select with LLMs(I)
if Name is sufficient **then**
  return v_2.ID
else if Attributes are sufficient **then**
  return v_2.ID
else
  Use relations to finalize match and **return** v_2.ID
end if

A.4 The algorithm of EasyEA

The algorithm flow is shown in Table 1.

A.5 Comparative Experiments of Different Feature Fusion Methods

Table 11 shows the performance of different fusion methods on the $\text{DBP15K}_{\text{ZH-EN}}$ dataset. The Concatenation Fusion method outperformed others, achieving the highest Hits@1 (**0.997**) and MRR (**0.996**), indicating its effectiveness in preserving the full information from multiple embeddings. In comparison, Max Pooling Fusion and Mean Fusion showed slightly lower performance, with Hits@1 scores of 0.996, respectively.

The differences in performance can be attributed to the characteristics of each fusion method. Max Pooling selects the maximum value from each embedding, which may overlook finer details, while

Mean Fusion averages the embeddings, potentially losing important features. Given its superior performance, Concatenation Fusion is chosen as the preferred method for candidate selection, as it provides the most detailed and comprehensive representation of embeddings, which is critical for high-precision EA.

This ablation experiment focuses on the feature fusion methods applied to the embeddings generated in Stage 1, and therefore does not include the Candidate Selection process from Stage 3. The primary aim is to evaluate the impact of different fusion strategies on the quality of embeddings, without considering the influence of subsequent candidate selection.

Settings	$\text{DBP15K}_{\text{ZH-EN}}$		
	Hits@1	Hits@10	MRR
EasyEA (Concatenation)	**0.997**	**1.000**	**0.996**
Max Pooling Fusion	0.996	1.000	0.995
Mean Fusion	0.996	0.999	0.995

Table 11: Performance results for different fusion methods on $\text{DBP15K}_{\text{ZH-EN}}$

A.6 Comparative Experiments of Candidate Selection with Different LLMs

In this experiment, we evaluated the performance of different LLMs (GPT-4o, GPT-4-Turbo, GPT-3.5-Turbo, and Llama3-70B) on reasoning tasks using a hierarchical strategy. The results in Table 12 show that all models achieved high performance, with GPT-4-turbo reaching the best result at 0.997, while the others (GPT-3.5-turbo, GPT-4o, and Llama3-70B) are similarly strong (0.996).

The results highlight the robustness of the hierarchical strategy across different LLMs. There are minor performance differences, and all models handle the reasoning tasks effectively. The consistency of results across various model architectures suggests that the strategy is highly generalizable and adaptable, making it a reliable approach for EA tasks with different LLMs.

Settings	Hits@1 of $\text{DBP15K}_{\text{ZH-EN}}$			
	GPT-4o	GPT-4-Turbo	GPT-3.5-Turbo	Llama3-70B
GPT-3.5+LLama3-8B	0.996	**0.997**	0.996	0.996

Table 12: Comparative results of LLMs reasoning on $\text{DBP15K}_{\text{ZH-EN}}$

A.7 Prompts

The prompts for translation, summary, and reasoning are shown in Tables 15, 16, and 17, respectively.

Dataset	Language	Entities	Relations	Attributes	Rel. Triples	Attr. Triples
DBP15K$_{ZH-EN}$	ZH	19,388	1,701	8,113	70,414	379,684
	EN	19,572	1,323	7,173	95,142	567,755
DBP15K$_{JA-EN}$	JA	19,814	1,299	5,882	77,214	354,619
	EN	19,780	1,153	6,066	93,484	497,230
DBP15K$_{FR-EN}$	FR	19,661	903	4,547	105,998	354,619
	EN	19,993	1,208	6,422	115,722	497,230
SRPRS$_{EN-FR}$	EN	15,000	221	296	36,508	70,750
	FR	15,000	177	415	33,532	56,344
SRPRS$_{EN-DE}$	EN	15,000	222	296	38,363	62,715
	DE	15,000	120	193	37,377	142,506
SRPRS$_{DBP-WIKI}$	DBpedia	15,000	253	363	38,421	71,957
	Wikipedia	15,000	144	652	40,159	136,315
SRPRS$_{DBP-YAGO}$	DBpedia	15,000	223	320	33,748	69,355
	YAGO3	15,000	30	22	36,569	22,519
DBP-WD	DBpedia	100,000	330	351	463,294	381,166
	Wikipedia	100,000	220	729	448,736	789,815
DBP-YG	DBpedia	100,000	302	334	428,952	451,646
	YAGO	100,000	31	23	502,563	118,376

Table 13: Statistical data of DBP15K, SRPRS and DWY.

Dataset	Entities	Relations	Facts	Density	Anchors	Overlapping	Struc. Sim.	Temporal
ICEWS-WIKI	11,047	272	3,527,881	319.352	5,058	45.79%	15.4%	Yes
	15,896	226	198,257	12.472		31.82%		Yes
ICEWS-YAGO	26,863	272	4,192,555	156.072	18,824	70.07%	14.0%	Yes
	22,734	41	107,118	4.712		82.80%		Yes

Table 14: Statistical data of ICEWS.

Translating Prompt
prompt = """ Translate the following entity names into English. You must remember that you can only give me the English entity name and cannot return any additional information. """

Table 15: Translating the name of entity into English

Summary Prompt

prompt = """
You are an expert who can provide concise explanations based on entity information.
I will give you the properties of an entity in the form of triples (subject, predicate, object).
Using this information along with your general knowledge,
please provide a short description of the entity.

- The explanation should be no longer than 100 words.
- Focus on summarizing the entity based on the given information and your general knowledge.
- Do not include unnecessary details or explanations beyond the entity description.

Example:
Entity Information: (Albert Einstein, profession, Physicist),
(Albert Einstein, known for, Theory of Relativity)
Explanation: Albert Einstein was a renowned physicist best known for developing
the Theory of Relativity, a fundamental theory in modern physics.

Now, please summarize the following entity information
and return a description in English:
"""

Table 16: Summarize entity information

Reasoning Prompt

prompt = """
I will provide you with a source entity and 10 target entities.
Your task is to select the target entity that most closely matches the source entity.

Each entity has three types of information:
1. Name information
2. Attribute triples
3. Relation triples

Follow this selection process:
1. Prioritize Name information as the primary criterion.
2. If Name information is ambiguous, use Attribute triples as a secondary criterion.
3. Finally, use Relation triples as the tertiary criterion.

Once you are confident, return only the ID of the target entity you believe is the best match.
Do not include any explanations, names, or other content in your response—ONLY the ID.
"""

Table 17: LLM selects the most likely matching entity