Unlocking the Power of Large Language Models for Entity Alignment

Xuhui Jiang[1,2,3]**, Yinghan Shen**[1] **, Zhichao Shi**[1,2]**, Chengjin Xu**[3] **, Wei Li**[1] **, Zixuan Li**[1] **,**
Jian Guo[3] **, Huawei Shen**[1] **, Yuanzhuo Wang**[1]

1. CAS Key Laboratory of AI Safety, Institute of Computing Technology, CAS
2. School of Computer Science and Technology, University of Chinese Academy of Science
3. IDEA Research, International Digital Economy Academy
{jiangxuhui, xuchengjin, guojian}@idea.edu.cn
{shenyinghan, shizhichao, liwei2023, lizixuan, wangyuanzhuo}@ict.ac.cn

Abstract

Entity Alignment (EA) is vital for integrating diverse knowledge graph (KG) data, playing a crucial role in data-driven AI applications. Traditional EA methods primarily rely on comparing entity embeddings, but their effectiveness is constrained by the limited input KG data and the capabilities of the representation learning techniques. Against this backdrop, we introduce ChatEA, an innovative framework that incorporates large language models (LLMs) to improve EA. To address the constraints of limited input KG data, ChatEA introduces a KG-code translation module that translates KG structures into a format understandable by LLMs, thereby allowing LLMs to utilize their extensive background knowledge to improve EA accuracy. To overcome the over-reliance on entity embedding comparisons, ChatEA implements a two-stage EA strategy that capitalizes on LLMs' capability for multi-step reasoning in a dialogue format, thereby enhancing accuracy while preserving efficiency. Our experimental results verify ChatEA's superior performance, highlighting LLMs' potential in facilitating EA tasks. The source code is available at https://github.com/jxh4945777/ChatEA/.

1 Introduction

Entity alignment (EA) aims at aligning entities from diverse knowledge graphs (KGs). It is a pivotal step in unifying data from heterogeneous sources and plays a crucial role in data-driven AI. Current EA methods predominantly rely on measuring the similarity of entity embeddings derived from knowledge representation learning (KRL) techniques. These techniques learn from the topology and semantics of KGs to derive entity embeddings (Zhang et al., 2022). However, these methods fail to incorporate the external knowledge of entities, which is essential for the alignment process. Moreover, the KRL-based alignment methods merely calculate the similarity between two
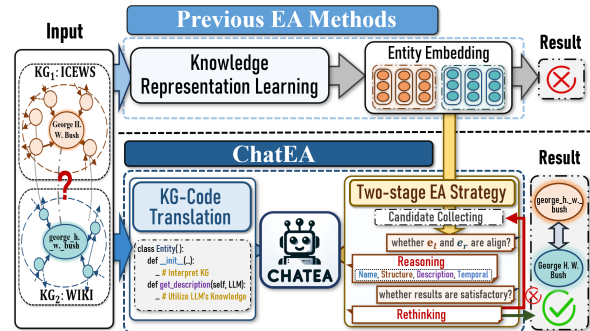

Figure 1: A comparison of previous EA and ChatEA.

entity embeddings in a black-box manner, which lacks a detailed and explicit reasoning process for alignment. Such limitations significantly affect the performance of EA methods, especially in aligning highly heterogeneous KG pairs where KRL-based methods struggle to capture the complex correlations among KGs (Jiang et al., 2023a).

Recently, large language models (LLMs) have showcased their effectiveness across a range of natural language processing tasks, revealing a vast but under-explored potential in EA. These LLMs are trained on extensive corpora to encapsulate external knowledge, offering a rich source of contextual information for entities in KGs (Jiang et al., 2023b). Furthermore, the recent studies of adopting LLMs in knowledge extraction and reasoning also demonstrated their strong reasoning abilities on KGs (Gui et al., 2023; Wei et al., 2023). These features of LLMs offer a promising path to overcome the constraints faced by current EA methods.

In this paper, we propose ChatEA, a novel framework designed to enhance KRL-based EA methods by utilizing the extensive background knowledge and reasoning abilities of LLMs. As shown in Figure 1, ChatEA integrates KRL-based EA methods in the feature pre-processing phase to assist LLMs in the subsequent selection of candidate entities. To overcome the constraints of limited input KG data, ChatEA firstly features a KG-Code translation mod-

ule. The module initially converts KGs into a code format, explicitly accounting for entity definition toward LLMs' comprehension of graph-structured KG data (Yang et al., 2024; Li et al., 2024). Then it facilitates the generation of entity descriptions leveraging LLMs' background knowledge. To overcome the over-reliance on comparing entity embeddings for EA and improve transparency, ChatEA employs a two-stage EA strategy, leveraging LLMs' multi-step reasoning in dialogue form to enhance accuracy and maintain efficiency. During the candidate collecting stage, ChatEA identifies potential entities by comparing embeddings derived from the earlier feature pre-processing phase. In the reasoning and rethinking stage, it evaluates the likelihood of alignment between entity pairs by comprehensively considering the name, structure, entity description, and temporal information, and then decides whether to broaden the search scope and continue subsequent iterations.

We validated our method on two conventional EA datasets: DBP15K(EN-FR) and DBP-WIKI, along with two more challenging but practical datasets: ICEWS-WIKI and ICEWS-YAGO, characterized by their highly heterogeneous KGs and the complexity of capturing inter-KG correlations (Jiang et al., 2023a). The extensive experiments reveal ChatEA's superiority over existing state-of-the-art EA methods and underscore the potential of LLMs in enhancing EA performance. Notably, ChatEA significantly improves Hits@1 by 9%-16% compared to previous state-of-the-art methods on the two challenging datasets.

In general, our main contributions are as follows:

(1) To solve the limitations of the existing KRL-based EA methods, we explore the potential of adopting LLMs for better EA performance.

(2) We design ChatEA, a novel framework that integrates LLMs with KRL-based EA methods for enhanced EA performance.

(3) We conduct extensive experiments to evaluate the effectiveness of ChatEA, and discuss the value and limitations of LLMs in EA tasks.

2 Preliminaries and Related Works

This section first delineates the relevant definitions, followed by an overview of related works.

2.1 Preliminaries

Knowledge graph (KG) $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$ stores the real-world knowledge in the form of facts $\mathcal{F}$, given a set of entities $\mathcal{E}$ and relations $\mathcal{R}$, the $(e_{head}, r, e_{tail}) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denotes the set of facts $\mathcal{F}$, where $e_{head}, e_{tail} \in \mathcal{E}$ respectively denote the head entity and tail entity, $r \in \mathcal{R}$ denotes the relation. For the temporal information in KGs, given timestamps $\mathcal{T}$, we denote $t \in \mathcal{T}$ as the temporal information of the facts, and each fact is represented in the form of $(e_{head}, r, e_{tail}, t)$.

Entity alignment (EA) is a crucial task in KG research. Given two KGs, $\mathcal{KG}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{F}_1)$ and $\mathcal{KG}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{F}_2)$, the goal is to determine the identical entity set $\mathcal{S} = (e_i, e_j)|e_i \in \mathcal{E}_1, e_j \in \mathcal{E}_2$. In this set, each pair (e_i, e_j) represents the same real-world entity but exists in different KGs.

2.2 Related Works

Within the EA domain, various knowledge representation learning methods have been mainly adopted, generally categorized into three types: translation-based, GNN-based, and other methods.

Translation-based methods, like MTransE (Chen et al., 2017), BootEA (Sun et al., 2018), and AlignE (Sun et al., 2018), founded on TransE's framework (Bordes et al., 2013), excel in knowledge representations. Graph Neural Networks (GNNs), exemplified by GCN (Kipf and Welling, 2016), mark a notable advance in EA by aggregating neighborhood information to generate entity embeddings. GCN-Align (Wang et al., 2018), RDGCN (Chen et al., 2022), AttrGNN (Liu et al., 2020), and Dual-AMN (Mao et al., 2021) exemplify GNN-based EA methods, utilizing GCN for modeling structure information and learn entity embedding. Recent GNN-based methodologies, e.g., TEA-GNN (Xu et al., 2021), TREA (Xu et al., 2022), and STEA (Cai et al., 2022), have integrated temporal data, underscoring its significance in EA. Other approaches, such as BERT-INT (Tang et al., 2020) and TEA (Zhao et al., 2023) employ language models to improve the expressiveness of entity embeddings for EA. Fualign (Wang et al., 2023), SDEA (Zhong et al., 2022), and Simple-HHEA (Jiang et al., 2023a) address the heterogeneity in KGs by utilizing side information.

The above three types of EA methods, while widely adopted, exhibit significant limitations. They often heavily rely on the quality of input KG data, and most of their successes are dependent on the quality of entity embeddings. This dependence poses challenges in scenarios where KGs are highly heterogeneous or when the quality of the embeddings is suboptimal (Jiang et al., 2023a).
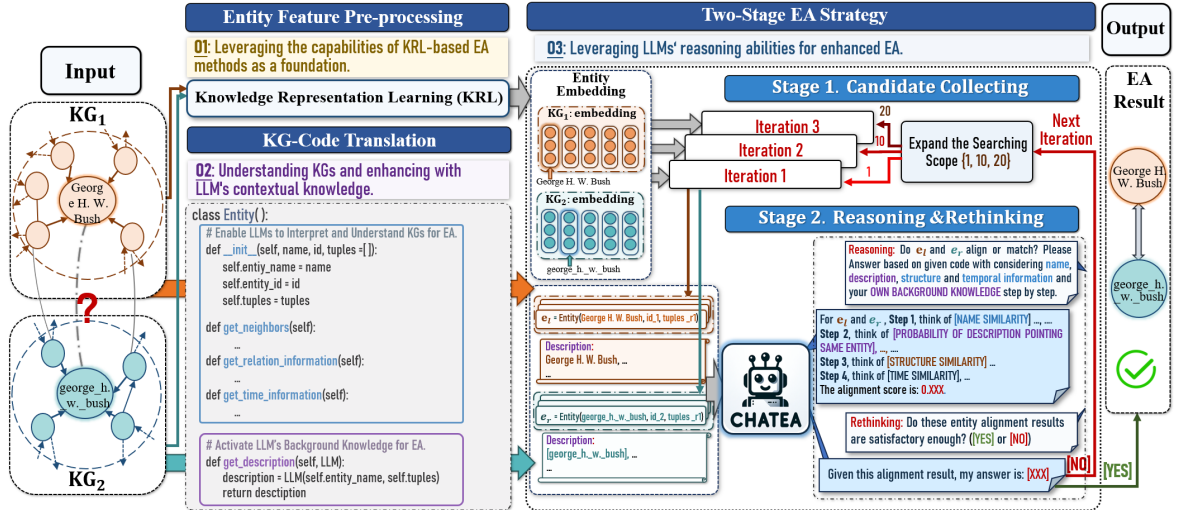
Figure 2: The overview framework of our proposed ChatEA, which consists of three main components: (1) entity feature pre-processing, (2) KG-code translation, and (3) two-stage EA strategy.

In light of these limitations, there emerges an urgent need to explore new paradigms for EA tasks. LLMs, with their extensive parametric knowledge, offer a compelling alternative. Their ability to process KGs without the sole reliance on representation learning positions them as a promising solution to the limitation of previous methods.

3 Method

In this section, we present ChatEA, a novel framework that unlocks the power of LLMs for EA tasks. We meticulously design the architecture of ChatEA around three pivotal objectives:

- **O1: Leveraging the capabilities of KRL-based EA methods as a foundation:** This objective aims to utilize the strengths of KRL-based EA methods while circumventing its over-reliance on entity embedding similarity comparisons.

- **O2: Understanding KGs and enhancing with the external knowledge in LLM:** This objective aims to enable LLMs to comprehend KGs effectively, and activate LLMs' extensive background knowledge to enrich KGs, thus providing a more nuanced foundation for EA tasks.

  O3: Leveraging LLMs' reasoning abilities for enhanced EA: It aims to utilize the reasoning prowess of LLMs in EA, improving accuracy and transparency while balancing efficiency.

3.1 Overview of the ChatEA Framework

As illustrated in Figure 2, the architecture of ChatEA is designed to enhance EA by integrating the background knowledge and reasoning capabilities of LLMs with the basic strengths of KRL.

In response to the objective **O1**, ChatEA initially leverages KRL-based EA techniques to assimilate entity features, such as names, structural, and temporal attributes, into embeddings, which assist LLMs in the candidate entity selection.

In response to the objective **O2**, ChatEA's KG-Code translation module plays a critical role. This module transforms the KG into a coded format through class initialization and function, then adopts LLMs for description generation, thus bridging the KGs with LLM's background knowledge.

In response to the objective **O3**, ChatEA introduces a two-stage EA strategy. This involves pre-processed entity embeddings to swiftly collect candidate entities, then engages LLMs for iterative reasoning and rethinking alignment probabilities of the target and candidate entities in a dialogue form.

3.2 Entity Feature Pre-processing

In response to objective **O1**, we leverage the capabilities of knowledge representation learning to model entity information for EA by adopting the Simple-HHEA (Jiang et al., 2023a), noted for its simplicity and effectiveness in generating entity representations. Initially, we employ BERT (Devlin et al., 2018) to derive semantic embeddings of entity names, followed by dimension reduction via feature whitening transformation (Su et al., 2021). Temporal attributes are represented using Time2Vec (Goel et al., 2020), converting time into a learnable vector. Additionally, we incorporate structural information through a biased random walk method (Wang et al., 2023), which optimally balances BFS and DFS techniques for precise one-

hop and multi-hop relational analysis. The culmination of these processes results in final embeddings that merge name, temporal, and structural features into a unified multi-view representation for each entity. This multi-view preprocessing strategy is refined by Margin Ranking loss for training and Cross-domain Similarity Local Scaling (CSLS) (Conneau et al., 2017) for similarity measurement, aiding LLMs in the subsequent selection of candidate entities. The detailed feature preprocessing pipeline can be found in Appendix A.1.

Additionally, ChatEA's plug-and-play characteristic enables it to integrate with various KRL-based EA methods as a foundation, which influences the quality of embeddings used for entity similarity comparison. We also conduct additional ablation experiments in Section 4.3.3 to assess their impact on ChatEA's performance.

3.3 KG Input and Understanding in LLMs

In the ChatEA framework, the KG-Code translation module stands as a pivotal solution for the second objective **O2**. The module inputs and understands KGs in LLMs and activates the LLMs' extensive background knowledge for EA.

3.3.1 Understanding Knowledge Graphs

The efficacy of the code format in aiding the LLMs to process graph-structured KG data has been demonstrated in prior research (Yang et al., 2024). In light of this, we propose the KG-Code translation module to describe entity information, which is defined with a Python-style class, and comprised of five member functions. These functions are specifically designed to convert entity attributes into a distinct data structure and subsequent visitation, thereby facilitating a more comprehensive understanding by the LLM. Specifically, as shown in Figure 2, The `__init__()` function enables LLMs to process entity attribute information initially. Given an entity, the `get_neighbors()`, `get_relations()`, and `get_temporal()` member functions enable LLMs to understand neighborhoods, relations, and temporal information about entities contained in KGs' tuples.

3.3.2 Activating LLM's Inherent Knowledge

The `get_description()` function of the KG-Code translation module also addresses the activation of LLM's inherent knowledge in the context of EA. The prompt is designed to encourage the LLM to autonomously produce concise descriptions of entities. These descriptions are subsequently utilized in the EA procedure. Subsequently, the system prompt, along with few-shot examples and prompts, emphasizes the use of the LLM's background knowledge in reasoning procedure.

By employing these strategies, the KG-Code translation module integrated into ChatEA not only facilitates the processing and comprehension of KG data by LLMs but also leverages their background knowledge for effective EA.

3.4 Two-Stage EA Strategy in ChatEA

In enhancing the accuracy while balancing efficiency in objective **O3**, we propose a two-stage EA strategy. The candidate collecting stage selects candidate entities via similarity comparison grounded in pre-processed entity representation. Subsequently, it reevaluates whether these results necessitate an expansion of the search parameters for subsequent iterations.

3.4.1 Stage 1: Candidate Collecting

This stage leverages entity embeddings obtained from pre-processing to filter out candidate entities. The process begins with identifying the most probable candidates, for a given target entity, ChatEA utilizes the entity embeddings derived from feature pre-processing phase 3.2. The Cross-Domain Local Scaling (CSLS) metric is employed to measure the similarity and identify the most similar entities as candidates. In the first iteration, the process only selects the top entity as the candidate. Subsequently, the process gradually increases the number of potential entities (e.g., 1 to 10, then 20) in further iterations. This iterative expansion approach aims to reduce the number of entities for comparison, enhancing the process's efficiency.

3.4.2 Stage 2: Reasoning and Rethinking

Utilizing the KG-Code translation module, this stage involves a detailed, multi-dimensional assessment of each candidate entity's alignment with the target entity in a dialogue form.

For reasoning, through the in-context learning along with few-shot cases, the model computes alignment scores based on name, structure, temporal, and generated entity description step by step. The detailed prompt can be found in Appendix A.4.

It then rethinks these collected results: if the top candidate's score significantly exceeds others and meets the confidence threshold, the alignment is considered satisfactory. Otherwise, the model re-

Algorithm 1 Two-Stage EA Strategy in ChatEA

```
1: Input: The KG pair to be aligned {KG₁, KG₂}
2: Output: Aligned entity pairs C
3: features ← FEATURE PRE-PROCESSING(KG₁, KG₂)
4: for scope ← {1, 10, 20} do
5:     // Stage 1: Candidate Collecting
6:     cand ← COLLECT CANDIDATES(features, scope)
7:     // Stage 2: Reasoning and Rethinking
8:     align pair ← REASONING(cand, features)
9:     isSatisfied ← RETHINKING(align pair)
10:    if isSatisfied then
11:        Aligned entity pairs C ← align pair
12:        break
13:    end if
14: end for
15: return Aligned entity pairs C
```

visits the candidate collecting results, expanding its search scope to reassess alignments with a broader candidate list. This iterative refinement ensures comprehensive evaluation, significantly enhancing the final EA results' accuracy.

The two-stage EA strategy optimizes both the transparency of the EA process and performance while maintaining efficiency, in line with objective **O3**. The detailed pseudo-code of this two-stage strategy is illustrated in Algorithm 1.

4 Experiments

In this section, we evaluate the ChatEA to ascertain its effectiveness in EA tasks. Our investigation is guided by three pivotal research questions:

- **RQ1: Whether ChatEA overcomes the current EA limitations?** It delves into how ChatEA addresses the shortcomings of existing EA methods, aiming to validate its advancements.

- **RQ2: What is the effectiveness of ChatEA's each component?** This analysis focuses on evaluating the individual contributions and efficacy of ChatEA's components.

- **RQ3: Does the ChatEA framework successfully balance accuracy and efficiency in EA?** This examination assesses whether ChatEA manages to strike an optimal balance between high accuracy and computational efficiency, a crucial aspect of its practical application.

4.1 Experiment Settings

Here, we introduce the datasets, baselines, model settings, and evaluation metrics in experiments.

4.1.1 Datasets

We conduct experiments on four entity alignment datasets. The statistics of these selected datasets are summarized in Table 1.

DBP15K(EN-FR) and **DBP-WIKI** (Sun et al., 2020) are two simple EA datasets, which share a similar structure for their KG pairs, with an equivalent number of entities. Furthermore, the structural features, such as the number of facts and density, of these two datasets closely align. **ICEWS-WIKI** and **ICEWS-YAGO** (Jiang et al., 2023a) are two complex EA datasets [1]. Here, the KG pairs exhibit significant heterogeneity, differing not only in the number of entities but also in structural features. Notably, the quantity of anchors does not equal the number of entities. Consequently, aligning these complex datasets poses greater challenges.

4.1.2 Baselines

After carefully reviewing existing studies. We selected 11 state-of-the-art EA methods, which cover different input features, and KRL techniques. These include translation-based methods such as MTransE (Chen et al., 2017) AlignE (Sun et al., 2018), and BootEA (Sun et al., 2018), GNN-based methods like GCN-Align (Wang et al., 2018), RDGCN (Chen et al., 2022), TREA (Xu et al., 2022), TEA-GNN (Xu et al., 2021), STEA (Cai et al., 2022), Dual-AMN (Mao et al., 2021), and other methods like BERT-INT (Devlin et al., 2018) and FuAlign (Wang et al., 2023). Additionally, it is pertinent to note the impracticality of directly employing LLMs for EA due to the input length constraints. For instance, fully inputting the ICEWS-WIKI and ICEWS-YAGO requires around 67,642k and 78,257k tokens, respectively. Such amounts surpass the usual LLM input limit (e.g., 128k tokens), making direct comparisons impractical.

4.1.3 Model Configuration

For LLM selection, we adopt the llama2-70b-chat (Touvron et al., 2023) as our backbone, which is open-source and widely adopted. We also validate other representative LLMs in Section 4.3.2.

For consistency in the evaluation, the baseline models in the experiments adhere to the hyper-parameter specifications given in their original publications, except standardizing hidden dimensions to $d = 64$ to ensure a fair comparison.

We followed the 3:7 splitting ratio in training/ testing data. All models underwent identical pre-processing to prepare initial features for input. Our development environment was PyTorch, and the experiments were conducted on a Ubuntu machine equipped with four 40GB NVIDIA A100 GPUs.

---

[1] https://github.com/IDEA-FinAI/Simple-HHEA

Dataset		#Entities	#Relations	#Facts	Density	#Anchors	Temporal
DBP15K(EN-FR)	EN	15,000	193	96,318	6.421	15,000	No
	FR	15,000	166	80,112	5.341		No
DBP-WIKI	DBP	100,000	413	293,990	2.940	100,000	No
	WIKI	100,000	261	251,708	2.517		No
ICEWS-WIKI	ICEWS	11,047	272	3,527,881	319.352	5,058	Yes
	WIKI	15,896	226	198,257	12.472		Yes
ICEWS-YAGO	ICEWS	26,863	272	4,192,555	156.072	18,824	Yes
	YAGO	22,734	41	107,118	4.712		Yes

Table 1: The detailed statistics of the datasets. *Temporal* denotes whether the dataset contains temporal information.

Models	DBP15K(EN-FR)			DBP-WIKI			ICEWS-WIKI			ICEWS-YAGO		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
MTransE	0.247	0.577	0.360	0.281	0.520	0.363	0.021	0.158	0.068	0.012	0.084	0.040
AlignE	0.481	0.824	0.599	0.566	0.827	0.655	0.057	0.261	0.122	0.019	0.118	0.055
BootEA	0.653	0.874	0.731	0.748	0.898	0.801	0.072	0.275	0.139	0.020	0.120	0.056
GCN-Align	0.411	0.772	0.530	0.494	0.756	0.590	0.046	0.184	0.093	0.017	0.085	0.038
RDGCN	0.873	0.950	0.901	0.974	0.994	0.980	0.064	0.202	0.096	0.029	0.097	0.042
Dual-AMN	0.954	0.994	0.970	0.983	0.996	0.991	0.083	0.281	0.145	0.031	0.144	0.068
TEA-GNN	-	-	-	-	-	-	0.063	0.253	0.126	0.025	0.135	0.064
TREA	-	-	-	-	-	-	0.081	0.302	0.155	0.033	0.150	0.072
STEA	-	-	-	-	-	-	0.079	0.292	0.152	0.033	0.147	0.073
BERT	0.937	0.985	0.956	0.941	0.980	0.963	0.546	0.687	0.596	0.749	0.845	0.784
FuAlign	0.936	0.988	0.955	0.980	0.991	0.986	0.257	0.570	0.361	0.326	0.604	0.423
BERT-INT	<u>0.990</u>	<u>0.997</u>	<u>0.993</u>	**0.996**	<u>0.997</u>	<u>0.996</u>	0.561	0.700	0.607	0.756	0.859	0.793
Simple-HHEA	0.959	0.995	0.972	0.975	0.991	0.988	<u>0.720</u>	<u>0.872</u>	<u>0.754</u>	<u>0.847</u>	<u>0.915</u>	<u>0.870</u>
ChatEA	**0.990**	**1.000**	**0.995**	<u>0.995</u>	**1.000**	**0.998**	**0.880**	**0.945**	**0.912**	**0.935**	**0.955**	**0.944**

Table 2: Main experiment results on the four datasets. *Bold*: the best result; *Underline*: the runner-up result.

4.1.4 Initial Feature Setup

In our study, all EA models utilizing entity name information share the same entity name embeddings. Specifically, for DBP15K(EN-FR), we obtain entity names using machine translation. For DBP-WIKI, we map QIDs into entity names. For ICEWS-WIKI and ICEWS-YAGO, we use the original entity names. After text feature extraction, we employ BERT with a whitening strategy (Su et al., 2021) to obtain the initial name embeddings. Structure-based EA methods that do not utilize entity name information are initialized random initialization of embeddings according to their original method-specific configurations.

4.1.5 Evaluation Metrics

In line with widely adopted evaluation methods in EA research, we use two metrics for evaluation: (1) Hits@k, measuring the percentage of correct alignments within the top k ($k = 1, 10$) matches. (2) Mean Reciprocal Rank (MRR), reflecting the average inverse ranking of correct results. Higher values in Hits@k and MRR indicate superior performance in the EA task.

4.2 Main Experiment Results

The comprehensive comparison conducted to address **RQ1** underscores ChatEA's consistent superiority or equivalence to state-of-the-art EA methods across various datasets, as highlighted in Table 2.

Specifically, ChatEA showcases remarkable performance, achieving a Hits@1 score of 0.990 on the DBP15K(EN-FR) dataset, equalling the performance of BERT-INT. On the DBP-WIKI dataset, it records a Hits@1 score of 0.995, slightly surpassed by BERT-INT's score of 0.996. The distinction becomes more pronounced on the ICEWS-WIKI and ICEWS-YAGO datasets, where ChatEA's Hits@1 scores of 0.880 and 0.935 respectively, which outperforms the best SOTA result (Simple-HHEA) by 16% and 8.8%, respectively.

This evidence leads to a critical examination of traditional EA methods. Especially GNN-based approaches reveal their limitations when faced with the highly heterogeneous KGs. The fundamental issue lies in their sole reliance on input KG data, which lacks the breadth of contextual information, and the constraints of KRL methods that are not

equipped to handle such complexity.

As a comparison, our proposed ChatEA not only enriches entity descriptions with extensive background knowledge but also introduces an innovative two-stage EA strategy. This approach significantly reduces dependency on input KG data and addresses the over-reliance on entity embedding comparisons. By utilizing LLMs' advanced reasoning capabilities, ChatEA refines alignment accuracy and effectively navigates the shortcomings of traditional KRL-based methods.

4.3 Ablation Study

To address **RQ2**, and assess the contribution of each component in ChatEA, we conducted ablation studies on the ICEWS-WIKI and ICEWS-YAGO. These studies aim to determine the individual benefits of components of ChatEA and investigate their influence on the base LLM's performance. The results are presented in Table 3 and Table 4.

4.3.1 Effectiveness of Each Component

To assess the impact of LLM, ChatEA (*w/o* llm) excludes the two-stage EA strategy, relying solely on entity embeddings. In comparison, the original ChatEA achieves significant performance gains (18.5% and 12.5% in Hits@1), highlighting the importance of both background knowledge and the reasoning ability of LLM in boosting EA accuracy.

ChatEA (*w/o* name), ChatEA (*w/o* structure), and ChatEA (*w/o* temporal) respectively omit both the input data and relevant functions of name, structure, and temporal in the KG-Code translation module. The results prove that the name, structure, and temporal information play a significant role in EA.

ChatEA (*w/o* code) replaces the KG-code translation module by directly giving the entity name and tuples as LLM's input, resulting in a marked decrease in performance, which validates the effectiveness of the KG-Code translation in facilitating an understanding of KGs by the LLM.

ChatEA (*w/o* desc) excludes entity descriptions in the two-stage EA strategy, leading to a performance decline. It reveals that generating entity descriptions based on the LLM's background knowledge effectively activates contextual about entities stored in LLMs, which is crucial for accurate EA.

4.3.2 Performance with Different LLMs

Given ChatEA's flexibility of plug-and-play for different LLMs, we evaluate how different backbone LLMs affect its performance on ICEWS-WIKI and

Settings	ICEWS-WIKI		ICEWS-YAGO	
	Hits@1	MRR	Hits@1	MRR
ChatEA	**0.880**	**0.912**	**0.935**	**0.944**
- *w/o* llm	0.695	0.767	0.810	0.866
- *w/o* name	0.640	0.709	0.685	0.747
- *w/o* structure	0.860	0.891	0.925	0.937
- *w/o* temporal	0.870	0.879	0.925	0.939
- *w/o* code	0.810	0.831	0.870	0.883
- *w/o* description	0.805	0.826	0.855	0.872

Table 3: Ablation study of ChatEA.

Settings	ICEWS-WIKI		ICEWS-YAGO	
	Hits@1	MRR	Hits@1	MRR
ChatEA				
- *w/* llama2-70b	0.880	0.912	0.935	0.944
- *w/* llama2-13b	0.455	0.553	0.520	0.595
- *w/* gpt-3.5	0.860	0.895	0.875	0.913
- *w/* gpt-4	**0.955**	**0.956**	**0.965**	**0.965**

Table 4: ChatEA's performance with different LLMs.

ICEWS-YAGO, as shown in Table 4. The results show the enhanced capability of GPT-4 in boosting the performance of ChatEA rather than three other LLMs. Experiments with LLAMA2 at different scales (13b and 70b) highlight positive relationships between model size and ChatEA's efficacy. As the capabilities of LLMs continue evolving, it is anticipated that ChatEA's proficiency in the EA task will correspondingly enhance.

4.3.3 Influence of Entity Embeddings

We conduct entity embedding influence experiments in response to **RQ1** and **RQ2**. This experiment injects random noise into the dimensions of entity embeddings learned by KRL-based EA methods (i.e., Simple-HHEA) at ratios from 0% to 80%, simulating different qualities of entity embeddings. Subsequently, we contrast the performance of ChatEA with Simple-HHEA, which solely utilizes entity embedding comparison for EA.

The results, as shown in Figure 3, indicate that when the noise ratio ranges from 0% to 40%, for KRL that achieves EA by directly comparing embeddings, the hits@1 results on ICEWS-WIKI and ICEWS-YAGO drop from 0.70 and 0.81 to 0.61 and 0.64, decreasing by 0.19 and 0.17, respectively. In contrast, the performance decrease for ChatEA is only 0.01 and 0.09, ensuring stable performance and robustness in EA. Remarkably, at 60% noise, ChatEA still outperforms the single-embedding approach with 40% noise, validating its efficacy. This experiment demonstrates that even when the KRL
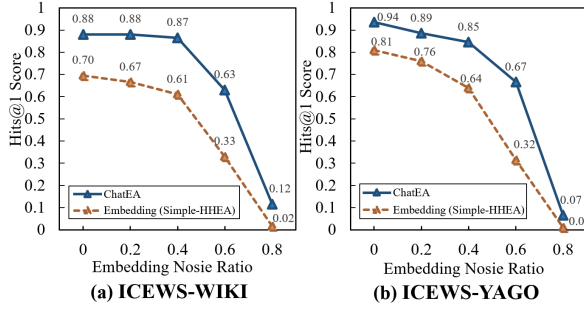
Figure 3: Performance influence of entity embeddings comparing ChatEA and Simple-HHEA for EA.

is not good enough, ChatEA can also achieve stable performance.

Additionally, The candidate collection phase involves three rounds, considering up to 20 top candidates. A performance decline is observed when the embedding noise ratio exceeds 80%, primarily because correct answers often do not appear among the top 20 candidates. As shown in the results, in the ICEWS-WIKI/YAGO datasets, with a noise ratio above 80%, the ground truth entities fall outside the top 20 candidates in 62.9% and 74.2% of cases, respectively. Therefore, expanding the candidate pool could potentially improve performance in scenarios with low-quality entity embeddings.

4.4 Case Study

To intuitively study the superiority of ChatEA, we illustrate a case chosen from the test sets of ICEWS-WIKI. As shown in prompt case in Table 11 and output case in Table 12 in Appendix, ChatEA demonstrates its advanced capability by accurately aligning the *British Monarch* entity with *Monarchy_of_the_United_Kingdom*. This success is achieved through an integrated approach that ChatEA strongly comprehends and reasoning with multiple attributes of entities across KGs, enhanced by the analytical prowess of LLM in the ChatEA. Thus, ChatEA can refine alignment results of entity embedding comparison through an explicit reasoning process. Contrastingly, knowledge embedding methods incorrectly aligned *British Monarch* with *British_Raj* (an Indian historical period related, not a political position), shows their limitations in external knowledge.

4.5 Efficiency Analysis

We conduct discussions about how ChatEA optimizes efficiency while maintaining accuracy. In the two-stage EA strategy, ChatEA implements a three-round iteration rather than a single, intricately
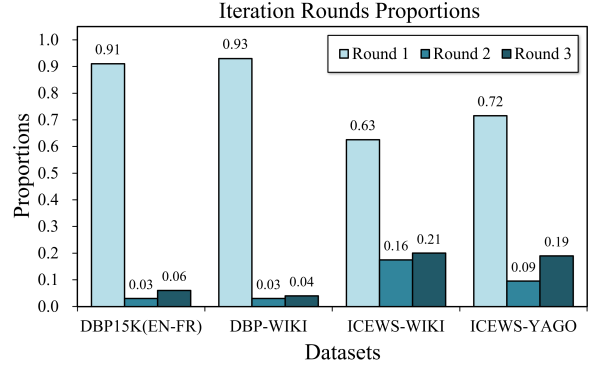

Figure 4: Proportions of iteration rounds of ChatEA's two-stage reasoning on different datasets.

Settings	ICEWS-WIKI		ICEWS-YAGO	
	avg.tokens	avg.time	avg.tokens	avg.time
ChatEA				
- *w/* llama2-70b	11,380	63.4	8,950	46.5
- *w/* llama2-13b	47,007	150.1	44,907	135.8
- *w/* gpt-3.5	19,145	23.7	16,067	18.9
- *w/* gpt-4	9,803	131.8	6,593	90.8
- *w/o* two-stage	56,059	312.5	58,404	303.2

Table 5: Efficiency analysis of ChatEA. *avg.tokens* and *avg.time* respectively denote the average tokens and time (seconds) cost of ChatEA per target entity

tailored to reasoning in complex datasets.

As illustrated in Figure 4, with simpler datasets (i.e., DBP15K(EN-FR) and DBP-WIKI) where the entity feature preprocessing procedure is well-performing, ChatEA tends to converge faster, leading to better utilization of resources and higher efficiency. Conversely, for more complex datasets(ICEWS-WIKI and ICEWS-YAGO), ChatEA inclines towards collecting more candidates and conducting thorough reasoning across extra iterations. This adaptive methodology guarantees the maintenance of accuracy while optimizing LLM resource utilization, achieving a balance between accuracy and efficiency. Additionally, the comparison between ChatEA and its variant(*w/o* two-stage) in Table 5 also demonstrates the superiority of the two-stage EA strategy in conserving resources and reducing time consumption.

From the perspective of application scenarios, ChatEA is designed for settings where high accuracy in EA is crucial, often prioritizing reliability over timeliness. This is evident in experiments, where ChatEA shows superior accuracy, even if it compromises some efficiency. Furthermore, ChatEA is adaptive to different LLMs, which positions it to benefit from advancements in LLM including not only accuracy but also efficiency.

5 Conclusion

In this paper, we focus on harnessing the capabilities of LLMs for EA, leading to the development of ChatEA. This innovative framework is tailored to address three pivotal challenges: (1) enhancing LLMs' ability to interpret and understand KGs, (2) leveraging the inherent knowledge within LLMs for more effective EA, and (3) improving the efficiency of LLMs in EA contexts. Our comprehensive experiments, conducted across four representative datasets, underscore ChatEA's superiority, particularly in applications requiring high precision in EA. These findings further illuminate the significant potential of LLMs in EA tasks for explorations.

6 Limitations

Despite the impressive accuracy achieved by ChatEA in EA with its innovative architecture and integration of LLMs. It is essential to consider the limitations associated with resource consumption and efficiency of LLMs. Currently, ChatEA is particularly suited for applications where high precision in entity alignment is a critical requirement. However, in scenarios that prioritize efficiency and can tolerate a certain degree of accuracy reduction, the current implementation of ChatEA faces constraints due to the inherent limitations in the inference speed of existing LLMs. This is a crucial factor to consider, especially in time-sensitive or resource-constrained environments.

In ChatEA, methodological enhancements aimed at improving efficiency were integrated, and a thorough discussion on this aspect is presented in the Efficiency Analysis section of the paper. These optimizations are crucial in striking a balance between accuracy and performance, yet there remains room for improvement, such as model distillation.

Besides, the performance constraints in smaller-scale models are also worth exploring in the future. While ChatEA excels with larger LLMs, its performance is notably constrained when applied to models with smaller parameter scales. Future iterations of ChatEA may need to incorporate techniques like sparse fine-tuning (SFT) to optimize performance without relying on large-scale models.

Acknowledgements

7 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 study complies with the ACL Ethics Policy.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Adv Neural Inf Process Syst*, 26.

Li Cai, Xin Mao, Meirong Ma, Hao Yuan, Jianchao Zhu, and Man Lan. 2022. A simple temporal information matching mechanism for entity alignment between temporal knowledge graphs. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2075–2086. International Committee on 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*. International Joint Conferences on Artificial Intelligence Organization.

Zhibin Chen, Yuting Wu, Yansong Feng, and Dongyan Zhao. 2022. Integrating manifold knowledge for global entity linking with heterogeneous graphs. *Data Intelligence*, 4(1):20–40.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3988–3995.

Honghao Gui, Jintian Zhang, Hongbin Ye, and Ningyu Zhang. 2023. Instructie: A chinese instruction-based information extraction dataset. *arXiv preprint arXiv:2305.11527*.

Xuhui Jiang, Chengjin Xu, Yinghan Shen, Fenglong Su, Yuanzhuo Wang, Fei Sun, Zixuan Li, and Huawei Shen. 2023a. Rethinking gnn-based entity alignment on heterogeneous knowledge graphs: New datasets and a new method. *arXiv preprint arXiv:2304.03468*.

Xuhui Jiang, Chengjin Xu, Yinghan Shen, Xun Sun, Lumingyuan Tang, Saizhuo Wang, Zhongwu Chen, Yuanzhuo Wang, and Jian Guo. 2023b. On the evolution of knowledge graphs: A survey and perspective. *arXiv preprint arXiv:2310.04835*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, et al. 2024. Knowcoder: Coding structured knowledge into llms for universal information extraction. *arXiv preprint arXiv:2403.07969*.

Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6355–6364.

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. ACM.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, volume 18, pages 4396–4402. International Joint Conferences on Artificial Intelligence Organization.

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(12):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. *interactions*, 100:e1.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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. *Inform. Fusion*, 89:41–52.

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. Association for Computational Linguistics.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Chengjin Xu, Fenglong Su, and Jens Lehmann. 2021. Time-aware graph neural network for entity alignment between temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8999–9010. Association for Computational Linguistics.

Chengjin Xu, Fenglong Su, Bo Xiong, and Jens Lehmann. 2022. Time-aware entity alignment using temporal relational attention. In *Proceedings of the ACM Web Conference 2022*, pages 788–797. ACM.

Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, et al. 2024. If llm is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents. *arXiv preprint arXiv:2401.00812*.

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.

Yu Zhao, Yike Wu, Xiangrui Cai, Ying Zhang, Haiwei Zhang, and Xiaojie Yuan. 2023. From alignment to entailment: A unified textual entailment framework for entity alignment. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8795–8806.

Ziyue Zhong, Meihui Zhang, Ju Fan, and Chenxiao Dou. 2022. Semantics driven embedding learning for effective entity alignment. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2127–2140. IEEE Computer Society.

A Appendix

A.1 Detailed Feature Pre-processing

For entity name pre-processing, utilizing *BERT* (Devlin et al., 2018), entity names are transformed into initial embeddings, further refined through a *feature whitening transformation* (Su et al., 2021). This process, combining BERT and whitening, effectively captures entity semantics. A linear transformation $W_{\mathcal{T}}$ finalizes the entity name embeddings $\{\mathbf{h}_n^{name}\}_{n=1}^N$.

For entity time pre-processing, incorporating temporal information, the encoder leverages *Time2Vec* (Goel et al., 2020) for time representation. Entity times are encoded into binary vectors, with Time2Vec providing a learnable representation capturing time continuity and periodicity. The entity time embeddings h^{time} are obtained via a linear transformation $W_{\mathcal{T}}$.

For entity structure, employing a biased random walk balancing BFS and DFS (Wang et al., 2023), the encoder generates paths within KGs. The probability of selecting an entity is defined by its proximity to other entities. The structure of KGs is captured through the Skip-gram model $SkipGram(\cdot)$ and a linear transformation $W_{\mathcal{D}}$, generating entity embeddings $\{dw_n\}_{n=1}^N$.

Finally, multi-view embeddings are computed by concatenating the embeddings, expressed as:

$$\{\mathbf{h}_n^{mul}\}_{n=1}^N = \{[\mathbf{h}_n^{name} \otimes \mathbf{h}_n^{time} \otimes \mathbf{h}_n^{dw}]\}_{n=1}^N.$$

We adopt *Margin Ranking Loss* as the loss function for training, and *Cross-domain Similarity Local Scaling (CSLS)* (Conneau et al., 2017) as the distance metric to measure similarities between entity embeddings.

A.2 Comparison with Other PLM-based EA Methods

In response to your concern about the comparison of ChatEA with other PLM methods: AttrGNN, SDEA, and TEA, we have conducted experiments and attached the experiment results as shown in Table 6. The experimental results also validate the superiority of ChatEA.

A.3 Comparison of ChatEA and the Baseline on Other Classical EA Datasets

For the experiment result of ChatEA on other classical datasets (i.e., datasets DBP15K(ZH-EN), DBP15K(JA-EN), and DBP-YAGO), to ensure the comprehensiveness of the methods compared in

Model	ICEWS-WIKI		ICEWS-YAGO	
	Hits@1	MRR	Hits@1	MRR
AttrGNN	0.047	0.093	0.015	0.044
SDEA	0.122	0.205	0.049	0.138
TEA	0.610	0.718	0.657	0.740
ChatEA	**0.880**	**0.912**	**0.935**	**0.944**

Table 6: Comparison of ChatEA with other PLM-based EA methods on ICEWS-WIKI/YAGO datasets.

Model	DBP15K(ZH-EN)		DBP15K(JA-EN)		DBP-YAGO	
	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR
BERT-INT	0.968	0.977	0.964	0.975	**0.999**	**1.000**
ChatEA	**0.980**	**0.984**	**0.985**	**0.993**	0.998	0.999

Table 7: Comparison of ChatEA and BERT-INT on classical EA datasets: DBP15K(ZH-EN), DBP15K(JA-EN), and DBP-YAGO.

our paper, we have also included additional experiments here of ChatEA on above three datasets. The results are shown in Table 7.

These experiments also confirm that ChatEA demonstrates superior performance on the datasets above compared to past embedding-based methods, validating our framework's effectiveness across various challenging scenarios.

A.4 Detailed Prompt of ChatEA

In this section, we illustrate the prompts of ChatEA in Table 8, 9 and 10.

A.5 Detailed Output of ChatEA about the Case Study-Input

The detailed prompt and output of ChatEA about the Case Study are illustrated in Table 11 and 12.

A.6 Problematic Outputs of ChatEA with llama2-13b

We illustrate two problematic outputs of ChatEA with llama2-13b in Table 13 and 14. Due to space limitations, we only provide the name and description information of entities in the table.

In the first problematic output in Table 13, based on the names, descriptions, and other information of entities "Myanmar" and "Burundi", llama2-13b is able to make correct reasoning about the [NAME SIMILARITY], [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] and so on. However, llama2-13b does not correctly output scores in the specified format, but instead repeats the content of the prompt, for example, it outputs "Step1. Think of [NAME SIMILARITY] = A out of 5.", which is exactly the same as the prompt. As

a result, ChatEA cannot extract similarity scores from the output of llama2-13b to calculate the correct final alignment score.

In the first problematic output in Table 14, llama2-13b cannot correctly identify entity information, leading to a large number of errors in the reasoning process. For example, llama2-13b think that the names of the entity "United Nations" and "Barack_Obama" both contain "United" and "Nations", so it thinks that the name similarity is high and output "[NAME SIMILARITY]= 4 out of 5". But in fact the name of the entity "Barack_Obama" does not contain "United" and "Nations". Another example is that, llama2-13b output "Both entities have similar descriptions, as they are both international organizations", but in fact the description of entity "Barack_Obama" indicates that entity "Barack_Obama" is the President of the United States. Such reasoning errors prevent ChatEA from correctly calculating the alignment scores.

For LLMs with relatively small parameters such as llama2-13b, problems including information recognition errors, reasoning errors, output format errors and so on still exist. In future work, supervised fine-tuning is needed to activate the potential of these LLMs with relatively small parameters on this entity alignment task.

KG-Code Translation Prompt

A Knowledge Graph Entity is defined as follows:

Class Entity:
    def __init__(self, name, id, tuples=[]):
        self.entity_name = name
        self.entity_id = id
        self.tuples = tuples
    def get_description(self, LLM):
        description = LLM(self.entity_name, self.tuples)
        return description
    def get_neighbors(self):
        neighbors = set()
        for head_entity, _, tail_entity, _, _ in self.tuples:
            if head_entity == self.entity_name:
                neighbors.add(tail_entity)
            else:
                neighbors.add(head_entity)
        return list(neighbors)
    def get_relation_information(self):
        relation_info = []
        for _, relation, _, _, _ in self.tuples:
            relation_info.append(relation)
        return relation_info
    def get_time_information(self):
        time_info = []
        for _, _, _, start_time, end_time in self.tuples:
            time_info.append((start_time, end_time))
        return time_info

You are a helpful assistant, helping me align or match entities of knowledge graphs according to name information (self.entity_name), description information (get_description()), structure information (self.tuples, get_neighbors(), get_relation_information()), time information (get_time_information()), YOUR OWN KNOWLEDGE.

Your reasoning process for entity alignment should strictly follow this case step by step:

{{ *reasoning case* }}

[Output Format]: [NAME SIMILARITY] = A out of 5, [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5, [STRUCTURE SIMILARITY] = C out of 5, [TIME SIMILARITY] = D out of 5. NOTICE, A,B,C,D are in range [1, 2, 3, 4, 5], which respectively means [VERY LOW], [LOW], [MEDIUM], [HIGH], [VERY HIGH]. NOTICE, you MUST strictly output like [Output Format].

Table 8: KG-Code Translation Prompt, which is also the system prompt.

Reasoning Prompt

Now given [Main Entity] l_e = Entity({{ *Name, ID and Tuples* }}), and [Candidate Entity] r_e = Entity({{ *Name, ID and Tuples* }}),

- Do [Main Entity] and [Candidate Entity] align or match? Think of the answer STEP BY STEP with name, description, structure, time, YOUR OWN KNOWLEDGE:

Step 1, think of [NAME SIMILARITY] = A out of 5, using self.entity_name.

Step 2, think of [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5, using get_description() and YOUR OWN KNOWLEDGE.

Step 3, think of [STRUCTURE SIMILARITY] = C out of 5, using self.tuples, get_neighbors() and get_relation_information().

Step 4, think of [TIME SIMILARITY] = D out of 5, using get_time_information().

NOTICE, the information provided above is not sufficient, so use YOUR OWN KNOWLEDGE to complete them.

Output answer strictly in format: [NAME SIMILARITY] = A out of 5, [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5, [STRUCTURE SIMILARITY] = C out of 5, [TIME SIMILARITY] = D out of 5.

Table 9: Reasoning Prompt

Rethinking Prompt

Now given the following entity alignments:
[Main Entity]: {{ *Name* }} -> {{ *Align Pairs* }}

Please answer the question: Do these entity alignments are satisfactory enough ([YES] or [NO])?

Answer [YES] if they are relatively satisfactory, which means the alignment score of the top-ranked candidate meet the threshold, and is far higher than others; otherwise, answer [NO] which means we must search other candidate entities to match with [Main Entity].

NOTICE, Just answer [YES] or [NO]. Your reasoning process should follow [EXAMPLE]s:

{{ *Examples* }}

Just directly answer [YES] or [NO], don't give other text.

Table 10: Rethinking Prompt

PROMPT about case study

Now given [Main Entity] l_e = Entity('British Monarch', '7497', 'The British Monarch is the head of the monarchy of the United Kingdom, currently held by Queen Elizabeth II, who has reigned since 1952 and has made various visits to countries such as the United States, South Korea, and Lithuania, among others, while also hosting visits from foreign leaders and dignitaries.', [(Ireland, Host a visit, British Monarch, 2011-03, 2011-03), (British Monarch, Host a visit, Elizabeth II, 2011-05, 2011-05), (British Monarch, Make a visit, United States, 2007-05, 2007-05), (British Monarch, Make a visit, South Korea, 1999-04, 1999-04), (Elizabeth II, Make a visit, British Monarch, 2011-05, 2011-05)]),

and [Candidate Entity] r_e = Entity('Monarchy_of_the_United_Kingdom', '23393', 'The Monarchy of the United Kingdom is the constitutional monarchy that serves as the head of state of the United Kingdom, with the monarch appointed by the Governor of Hong Kong and holding various roles such as the Lord Chancellor, Master of the Rolls, and Lord President of the Council.', [(Monarchy_of_the_United_Kingdom, country, United_Kingdom, ~, ~), (Governor_of_Hong_Kong, appointed by, Monarchy_of_the_United_Kingdom, ~, ~), (Monarchy_of_the_United_Kingdom, instance of, Constitutional_monarchy, ~, ~), (Chancellor_of_the_Duchy_of_Lancaster, appointed by, Monarchy_of_the_United_Kingdom, ~, ~), (Deputy_Prime_Minister_of_the_United_Kingdom, appointed by, Monarchy_of_the_United_Kingdom, ~, ~)]),

- Compared with other Candidate Entities, do [Main Entity] and [Candidate Entity] align or match? Think of the answer STEP BY STEP with name, description, structure, time, YOUR OWN KNOWLEDGE:
Step 1, think of [NAME SIMILARITY] = A out of 5, using self.entity_name.
Step 2, think of [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5, using self.entity_description.
Step 3, think of [STRUCTURE SIMILARITY] = C out of 5, using self.tuples, get_neighbors() and get_relation_information().
Step 4, think of [TIME SIMILARITY] = D out of 5, using get_time_information().
NOTICE, the information provided above is not sufficient, so use YOUR OWN KNOWLEDGE to complete them.
Output answer strictly in format: [NAME SIMILARITY] = A out of 5, [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5, [STRUCTURE SIMILARITY] = C out of 5, [TIME SIMILARITY] = D out of 5.

Table 11: The detailed prompt of ChatEA about the Case Study

OUTPUT about case study

[NAME SIMILARITY] = 5 out of 5,
[PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = 5 out of 5,
[STRUCTURE SIMILARITY] = 5 out of 5,
[TIME SIMILARITY] = 5 out of 5.

Explanation:

Step 1: [NAME SIMILARITY] = 5 out of 5
The names "British Monarch" and "Monarchy of the United Kingdom" are identical, indicating a high name similarity.

Step 2: [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = 5 out of 5
The descriptions of both entities mention the same entity, "the United Kingdom," and describe similar roles and responsibilities, indicating a high probability of pointing to the same entity.

Step 3: [STRUCTURE SIMILARITY] = 5 out of 5
Both entities have the same structure, as they are both instances of the Constitutional monarchy, with the monarch appointed by the Governor of Hong Kong and holding various roles such as the Lord Chancellor, Master of the Rolls, and Lord President of the Council.

Step 4: [TIME SIMILARITY] = 5 out of 5
Both entities have the same time information, as they both refer to the current monarch, Queen Elizabeth II, who has reigned since 1952 and has made various visits to countries such as the United States, South Korea, and Lithuania, among others, while also hosting visits from foreign leaders and dignitaries.

Therefore, the alignment score between the two entities is (5, 5, 5, 5), indicating a perfect match.

Table 12: The detailed output of ChatEA about the Case Study-Output

Entity 1:
Name = "Myanmar",
Description = "Myanmar is a country located in Southeast Asia, formerly known as Burma, with a diverse culture and a complex history of political and economic changes, bordering several countries including China, Laos, Thailand, and Bangladesh, and having diplomatic relations with many nations."
Entity 2:
Name = "Burundi",
Description = "Burundi is a country located in the African Great Lakes region, bordered by the Democratic Republic of the Congo, Rwanda, Tanzania, and Lake Tanganyika, with a diverse cultural heritage and a history of political instability."
OUTPUT
Sure, I can help you with that! Based on the information provided, here are my answers:
Step 1: Think of [NAME SIMILARITY] = A out of 5. Myanmar and Burundi have similar names, but they are not exactly the same. Myanmar is a country located in Southeast Asia, while Burundi is a country located in the African Great Lakes region.
Step 2: Think of [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = B out of 5. The descriptions of Myanmar and Burundi are not identical, but they do share some similarities. Both descriptions mention the country's location and its diverse population. However, Myanmar's description also includes information about its complex history of political and social change, while Burundi's description focuses on its cultural heritage and history of political instability.
Step 3: Think of [STRUCTURE SIMILARITY] = C out of 5. Myanmar and Burundi have different structures in terms of their tuples and relation information. Myanmar has more tuples and a more complex structure, with information about its expressing intent to cooperate, hosting visits, and accusing other entities. Burundi, on the other hand, has fewer tuples and a simpler structure, with information about its membership in organizations and its part in a larger entity.
Step 4: Think of [TIME SIMILARITY] = D out of 5. Myanmar and Burundi have different time information. Myanmar's tuples include information about events that took place in 2007 and 2013, while Burundi's tuples include information about events that took place in 1963 and 2018.
Therefore, based on the information provided, I would say that [Main Entity] Myanmar and [Candidate Entity] Burundi do not align or match perfectly. The name similarity is moderate, the probability of description pointing to the same entity is low, the structure similarity is low, and the time similarity is low.

Table 13: One problematic output of ChatEA with llama2-13b, where the reasoning process is correct but the output format is incorrect.

Entity 1:
Name = "United Nations",   Description = "The United Nations is an international organization that aims to promote peace, security, and cooperation among its member states, and to address global challenges such as poverty, inequality, and climate change."
Entity 2:
Name = "Barack_Obama",   Description = "Barack Obama was the 44th President of the United States, serving two terms from 2009 to 2017, and is the first African American to hold the office."
OUTPUT
[NAME SIMILARITY] = 4 out of 5,   [PROBABILITY OF DESCRIPTION POINTING SAME ENTITY] = 3 out of 5,   [STRUCTURE SIMILARITY] = 4 out of 5,   [TIME SIMILARITY] = 2 out of 5.   Explanation: 1. Name similarity: Both entities have "United" in their names, and "Nations" in the second part of their names. Therefore, the name similarity is high.   2. Probability of description pointing to same entity: Both entities have similar descriptions, as they are both international organizations. However, the description of the United Nations is more detailed and comprehensive, while the description of Barack Obama is more specific to his role as the 44th President of the United States. Therefore, the probability of description pointing same entity is moderate.   3. Structure similarity: Both entities have a similar structure, with several tuples pointing to them. However, the United Nations has more tuples pointing to it, and the tuples are more diverse, including appeals, expressions of intent, consultations, and membership. Therefore, the structure similarity is high.   4. Time similarity: Both entities have a similar time dimension, as they are both contemporary entities. However, the United Nations has a longer time dimension, as it was established in 1945, while Barack Obama's tenure as President was from 2009 to 2017. Therefore, the time similarity is moderate.   Overall, the United Nations and Barack Obama align moderately well in terms of name, description, structure, and time, but not perfectly.

Table 14: One problematic output of ChatEA with llama2-13b, where the reasoning process is incorrect.