Entity Alignment with Noisy Annotations from Large Language Models

Shengyuan Chen
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
shengyuan.chen@connect.polyu.hk

Qinggang Zhang
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
qinggangg.zhang@connect.polyu.hk

Junnan Dong
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
hanson.dong@connect.polyu.hk

Wen Hua
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
wency.hua@polyu.edu.hk

Qing Li
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
csqli@comp.polyu.edu.hk

Xiao Huang
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
xiaohuang@comp.polyu.edu.hk

Abstract

Entity alignment (EA) aims to merge two knowledge graphs (KGs) by identifying equivalent entity pairs. While existing methods heavily rely on human-generated labels, it is prohibitively expensive to incorporate cross-domain experts for annotation in real-world scenarios. The advent of Large Language Models (LLMs) presents new avenues for automating EA with annotations, inspired by their comprehensive capability to process semantic information. However, it is nontrivial to directly apply LLMs for EA since the annotation space in real-world KGs is large. LLMs could also generate noisy labels that may mislead the alignment. To this end, we propose a unified framework, LLM4EA, to effectively leverage LLMs for EA. Specifically, we design a novel active learning policy to significantly reduce the annotation space by prioritizing the most valuable entities based on the entire inter-KG and intra-KG structure. Moreover, we introduce an unsupervised label refiner to continuously enhance label accuracy through in-depth probabilistic reasoning. We iteratively optimize the policy based on the feedback from a base EA model. Extensive experiments demonstrate the advantages of LLM4EA on four benchmark datasets in terms of effectiveness, robustness, and efficiency.

1 Introduction

Knowledge graphs (KGs) serve as a foundational structure for storing and organizing structured knowledge about entities and their relationships, which facilitates effective and efficient search capabilities across various applications. They have been widely applied in question-answering systems (Dong et al., 2023, 2024c), recommendation systems (Catherine & Cohen, 2016; Chen et al., 2024a), social network analysis (Tang et al., 2008), Natural Language Processing (Weikum & Theobald, 2010), etc. Despite their extensive utility, real-world KGs often suffer from issues

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0482

such as incompleteness, domain specificity, or language constraints, which limit their effectiveness in cross-disciplinary or multilingual contexts. To address these challenges, entity alignment (EA) aims to merge disparate KGs into a unified, comprehensive knowledge base by identifying and linking equivalent entities across different KGs. For instance, by aligning entities between a financial KG and a legal KG, EA facilitates the understanding of complex relationships, such as identifying the same corporations across the two KGs to assess how legal regulations impact their financial performance. This alignment enables a more nuanced exploration and interrogation of interconnected data, providing richer insights into how entities operate across multiple domains.

Entity alignment models predict the equivalence of two entities by measuring their alignment probability. Specifically, rule-based methods (Suchanek et al., 2012; Jiménez-Ruiz & Cuenca Grau, 2011; Qi et al., 2021) utilize predefined rules or heuristics to update alignment probabilities and propagate alignment labels. Conversely, embedding-based models seek to exploit advanced techniques in graph learning (Li et al., 2024; Liu et al., 2024b,a, 2023), parameterizing these probabilities using similarity scores between entity representations learned through knowledge graph embedding algorithms such as translation models (Chen et al., 2017; Sun et al., 2018) or Graph Convolutional Networks (GCNs) (Wu et al., 2019; Mao et al., 2021; Wang et al., 2018; Huang et al., 2023). However, these methods heavily rely on extensive and accurate seed alignments for training—a requirement that poses significant challenges. The need for substantial, cross-domain knowledge to annotate such alignments often makes their acquisition prohibitively expensive.

Recently, Large Language Models (LLMs) have showcased their superior capability in processing semantic information Dong et al. (2024a), which has significantly advanced various graph learning tasks such as node classification (Chen et al., 2024d), graph reasoning (Zhao et al., 2023a; Chai et al., 2023), recommender systems (Zhou et al., 2022; Wu et al., 2023), SQL query generation (Zhang et al., 2024a), and knowledge graph-based question answering (Wang et al., 2024; Zhang et al., 2024b; Dong et al., 2024b). Their capacity to extract meaningful insights from graph data opens up new possibilities for automating EA. Notably, recent studies (Zhong et al., 2022; Zhao et al., 2023b; Jiang et al., 2024) have explored the use of LLMs in EA, primarily focusing on finetuning a pretrained LLM such as Bert to learn semantic-aware representations, relying on accurate seed alignments as training labels. Yet, the potential of LLMs for label-free EA via in-context learning remains unexplored.

However, directly applying LLMs to automate EA poses significant challenges. Firstly, conventional EA models presume that all annotations are correct; yet, LLMs can generate false labels due to LLMs' inherent randomness and the potential incompleteness or ambiguity in the semantic information of entities. Training an EA model directly on these noisy labels can severely impair the final alignment performance. Secondly, given the vast number of entity pairs, annotation with LLMs would be prohibitively expensive. Maximizing the utility of a limited LLM query budget is essential. Existing solutions such as active learning cannot be directly applied since the annotations are noisy.

In response to the outlined challenges, we introduce LLM4EA, a unified framework designed to effectively learn from noisy pseudo-labels generated by LLMs while dynamically optimizing the utility of a constrained query budget. LLM4EA actively selects source entities based on feedback from a base EA model, focusing on those that significantly reduce uncertainty for both the entities themselves and their neighbors. This approach allocates the query budget to important entities, guided by the intra-KG and inter-KG structure. To manage the noisy pseudo-labels effectively, LLM4EA incorporates an unsupervised label refiner that enhances label accuracy by selecting a subset of confident pseudo-labels through probabilistic reasoning. These refined labels are then utilized to train the base EA model for entity alignment. The confident alignment results inferred by the EA model inform active selection in subsequent iterations, thereby progressively improving the framework's effectiveness in a coherent and integrated manner. Contributions are summarized as follows:

- **Novel LLM-based framework for entity alignment:** We propose LLM4EA, an in-context learning framework that uses an LLM to annotate entity pairs. Leveraging the LLMs' zero-shot learning capability, this framework generates pseudo-labels, providing a foundation for entity alignment without ground truth labels.
- **Unsupervised label refinement:** Our framework introduces an unsupervised label refiner informed by probabilistic reasoning. This component significantly improves the accuracy of LLM-derived pseudo-labels, enabling effective training of entity alignment models.
- **Active sampling module:** We propose an active selection algorithm that dynamically searches entities in the huge annotation space. Guided by feedback from the EA model,

this algorithm adjusts its policy based on inter-KG and intra-KG structures, optimizing the utility of LLM queries and ensuring efficient use of resources.

- **Empirical validation and superior performance:** We rigorously evaluate our framework through extensive experiments and ablation studies. The results demonstrate that LLM4EA not only outperforms baselines by a large margin but also shows robustness and efficiency. Each component of the framework is shown to contribute meaningfully to the overall performance, with clear synergistic effects observed among the components.

2 Problem definition

A knowledge graph $\mathcal{G}$ comprises a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of relation triples $\mathcal{T}$ where each triple $(e_h, r, e_t) \in \mathcal{T}$ represents a directional relationship between its head entity and tail entity. Given two KGs $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, $\mathcal{G}' = \{\mathcal{E}', \mathcal{R}', \mathcal{T}'\}$ and a fixed query budget $\mathcal{B}$ to a Large Language Model, we aim to train an entity alignment model θ based on the LLM's annotations to infer the matching score $m_\theta(e, e')$ for all entity pairs $\{(e, e'), e \in \mathcal{E}, e' \in \mathcal{E}'\}$. The evaluation process utilizes a ground truth alignment set $\mathcal{A}$ to assess the prediction accuracy for target entities in both directions, i.e.,$(e, ?)$ and $(?, e')$ for each true pair $(e, e') \in \mathcal{A}$, based on the ranked matching scores m_θ. Evaluation metrics are hit@k (where $k \in \{1, 10\}$) and mean reciprocal rank (MRR).

3 Entity alignment with noisy annotations from LLMs

We aim to design a framework to perform entity alignment with LLMs. Our design is motivated by the following insights. Firstly, we have a huge search space (the overall annotation space is $O(|\mathcal{E}||\mathcal{E}'|)$) to identify the core entity pairs to annotate. Secondly, we don't know whether the annotated labels are correct or not, because we have no prior knowledge or heuristic of the label distribution. Finally, we perform annotations iteratively, requiring the model to adjust its search policy based on annotation effectiveness, while we have no verifiable feedback of this annotation accuracy.

Based on these insights, we propose LLM4EA—an iterative framework that consists of four interconnected steps in each cycle, as illustrated in Figure 1. Initially, **an active selection** technique optimizes the use of resources by choosing critical source entities that significantly reduce uncertainty for themselves and their neighbors. Subsequently, **an LLM-based annotator** identifies the counterparts for the selected source entities, generating a set of pseudo-labels. Next, **a label refiner** improves label accuracy by eliminating structurally incompatible labels. This process involves formulating a combinatorial optimization problem and utilizing a probabilistic-reasoning-based greedy search algorithm to efficiently find a local-optimal solution. Finally, these refined labels are used to train **a base EA model** for the entity alignment task. The outcomes of the alignment then serve as feedback to inform subsequent rounds of the active selection policy. Further details are provided below.

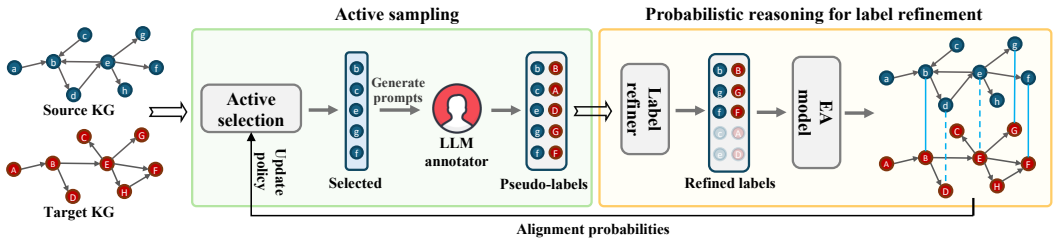

Figure 1: Overview of the LLM4EA framework. LLM4EA utilizes active sampling to select important entities based on feedback from an EA model. It also includes a label refiner to effectively train the base EA model using noisy pseudo-labels. Feedback from the EA model updates the selection policy.

3.1 Active selection of source entity

We aim to maximize the utility of the budget by actively allocating the budget to those beneficial entities. To do this, we sample source entities that reduce the most uncertainty of both themselves and their neighboring entities, by a dynamically adjusted policy. The measurement of uncertainty

https://doi.org/10.52202/079017-0482

reduction is based on two assumptions: 1) an entity's own uncertainty is inversely proportional to its alignment probability with its most probable counterpart; 2) the amount of uncertainty an entity eliminates for its neighbors is closely linked to the relational ties between them. To systematically assess this, we introduce the concept of *relational uncertainty*, quantified as follows:

$$U_r(e_h) = (1 - P(e_h)) + \sum_{(e_h,r,e_t)\in\mathcal{T}} w_r \left(1 - P(e_t)\right). \tag{1}$$

Here, w_r is a weight coefficient reflecting the significance of relation r and signifies how much e_h contributes to reducing the uncertainty of e_t through the relation r. For this purpose, we employ functionality $\mathcal{F}(r)$ (formally defined in Eq. (4)) as the weight w_r, as it quantifies the uniqueness of the tail entity for a given specified head entity. $P(e) \coloneqq \max_{e'\in\mathcal{E}'} P(e \equiv e')$ represents the alignment probability of the top-match entity for e. These alignment probabilities $P(e \equiv e')$ are obtained through probabilistic reasoning during label refinement (Section 3.3.2) and are augmented by the inferred alignments from the base EA model (Section 3.4). In the initial iteration, all alignment probabilities are set to 0.

It's important to note that some source entities are linked to a large number of uncertain neighbors (those with low $P(e)$). These source entities are crucial but may be overlooked if their connected relations have low functionality. Hence, we introduce *neighbor uncertainty* as another metric to assess an entity's importance, by removing the functionality-based weight coefficient:

$$U_n(e_h) = (1 - P(e_h)) + \sum_{(e_h,r,e_t)\in\mathcal{T}} \left(1 - P(e_t)\right). \tag{2}$$

To integrate these two metrics, we employ rank aggregation by mean reciprocal rank:

$$U(e_h) = 2 \times \left(\frac{1}{r_{ur}(e_h)} + \frac{1}{r_{un}(e_h)}\right). \tag{3}$$

Here, $r_{ur}(e_h)$ and $r_{un}(e_h)$ denote the ranking of e_h when using U_r and U_n as metric, respectively. This simple-effective aggregation technique is advantageous for our task since it's scale invariant and requires no validation set for tuning hyperparameters, making it more practical in this task.

3.2 LLM as annotator

Counterpart filtering. With the selected source entities, we employ an LLM as an annotator to identify the counterpart from $\mathcal{E}'$ for each source entity, generating a set of pseudo-labels $\mathcal{L} = \{(e, e')|e \in \mathcal{E}, e' \in \mathcal{E}'\}$. To narrow down the search space, we first filter out the less likely counterparts before querying the LLM, selecting only the top-k most similar counterparts from $\mathcal{E}'$. The similarity metric is flexible: we use a string matching score based on word edit distance, but other methods are also viable, such as semantic embedding distances derived from word embedding models. By adjusting k, we can trade-off between the recall rate of counterparts and the query cost.

Prompt design. There are primarily two methods for retrieving context information to construct textual prompts: randomly generated prompts and dynamically tuned prompts. The former involves randomly selecting neighbors to construct contexts for the entity, while the latter dynamically selects neighbors based on feedback from the EA model. For a fair comparison, we use randomly generated prompts across all baselines and the proposed LLM4EA. These prompts include the name of each entity and a set of relation triples to three randomly selected neighbors. For the baseline models, pseudo-labels are generated at once and used for training. For LLM4EA, we evenly divide the budget $\mathcal{B}$ into n iterations and generate pseudo-labels at each iteration using the allocated $\mathcal{B}/n$ budget.

3.3 Probabilistic reasoning for label refinement

The pseudo-labels generated by the LLM can be noisy, and directly using these labels to train an entity alignment (EA) model could undermine the final performance. Although estimating the label distribution by asking the LLM for confidence scores or querying multiple times to measure consistency are potential solutions, these approaches can be vulnerable or introduce additional costs.

In light of this, we propose a label refiner that leverages the structure of knowledge graphs. The refinement process is framed as a combinatorial optimization problem aimed at minimizing overall

structural incompatibility among labels. Utilizing a probabilistic reasoning technique, we progressively update our confidence estimation for each label and select those that are mutually compatible, ultimately producing a set of accurate pseudo-labels. Detailed explanations follow below.

3.3.1 Functionality and probabilistic reasoning

Functionality. The functionality of a relation quantifies the uniqueness of tail entities for a specified head entity, calculated as the ratio of unique head entities to total head-tail pairs linked by the relation. Conversely, inverse functionality quantifies the tail entity uniqueness for a specified head entity. Formally, these are defined as:

$$\mathcal{F}(r) \coloneqq \frac{|\{e_h|(e_h,r,e_t) \in \mathcal{T}\}}{|\{(e_h,e_t)|(e_h,r,e_t) \in \mathcal{T})\}|}, \quad \mathcal{F}^{-1}(r) \coloneqq \frac{|\{e_t|(e_h,r,e_t) \in \mathcal{T}\}}{|\{(e_h,e_t)|(e_h,r,e_t) \in \mathcal{T})\}|}. \tag{4}$$

For instance, suppose a KG contains two triples for the relation $locate\_in$: $(Hawaii, locate\_in, US)$ and $(Miami, locate\_in, US)$. Then $\mathcal{F}(locate\_in) = 1.0$ and $\mathcal{F}^{-1}(locate\_in) = 0.5$. In other words, given $(Miami, locate\_in, ?)$, the answer for the missing tail entity is unique; while given $(?, locate\_in, US)$, there are multiple answers for the missing head entity. Such relational patterns are useful for identifying an entity based on its connections within the intra-graph structure.

Probabilistic reasoning. If two entities are each connected to entities that are aligned across KGs, this increases the likelihood that they should be aligned as well. Based on this heuristic, an entity pair's alignment probability $P(e_h \equiv e'_h)$ can be inferred by aggregating its neighbors' alignment probability via relation functionality:

$$1 - \prod_{\substack{(e_h,r,e_t)\in\mathcal{T},\\(e'_h,r',e'_t)\in\mathcal{T}'}} \left(1 - \mathcal{F}^{-1}(r)P(r \subseteq r')P(e_t \equiv e'_t)\right) \times \left(1 - \mathcal{F}^{-1}(r')P(r' \subseteq r)P(e_t \equiv e'_t)\right). \tag{5}$$

Here, $P(r \subseteq r')$ denotes the probability of r being a subrelation of r', estimated by alignment probabilities of connected entities:

$$\frac{\sum\left(1 - \prod_{(e'_h,r',e'_t)\in\mathcal{T}'}\left(1 - P(e'_h \equiv e_h)P(e'_t \equiv e_t)\right)\right)}{\sum\left(1 - \prod_{e'_h,e'_t\in\mathcal{E}'}\left(1 - P(e'_h \equiv e_h)P(e'_t \equiv e_t)\right)\right)}. \tag{6}$$

These formulations allow for the propagation and updating of alignment probabilities in a manner that is cognizant of relational structures. We employ this technique to design a label refiner below.

3.3.2 Label refiner

Label incompatibility. We exploit the "incompatibility" of labels for label refinement, based on the assumption that correct labels can infer each other, while a false label could be incompatible with its correctly aligned neighbors. We define the *overall incompatibility* on a label set $\mathcal{L}$ as:

$$\Phi(\mathcal{L}) \coloneqq \sum_{(e_h,e'_h)\in\mathcal{L}} \left(\mathbf{1}_{P(e_h\equiv e'_h)<\max_{e\in\mathcal{E}} P(e,e'_h)} + \mathbf{1}_{P(e_h\equiv e'_h)<\max_{e'\in\mathcal{E}'} P(e_h,e')}\right). \tag{7}$$

Here, $\mathbf{1}_{P(e_h\equiv e'_h)<\max_{e\in\mathcal{E}} P(e,e'_h)} = 1$ if e_h is not the top-match for e'_h, otherwise 0. It's important to note that a detected incompatibility doesn't necessarily indicate the false alignment of (e_h, e'_h): it may suggest a misalignment of their neighbors. Given this, the key to label refinement is to jointly optimize the label's overall incompatibility while avoiding accidentally filtering out correct labels.

Objective. To enhance label quality, we propose to refine the pseudo-label set $\mathcal{L}$ by finding a subset $\mathcal{L}^* \subset \mathcal{L}$ that minimizes its overall incompatibility: $\mathcal{L}^* = \text{argmin}_{\mathcal{L}'\subset\mathcal{L}}\,\Phi(\mathcal{L}')$. Noteworthy that a trivial solution for this optimization problem is only preserving a set of isolated labels, such that $\max_{e\in\mathcal{E}} P(e \equiv e'_h) = 0$ and $\max_{e'\in\mathcal{E}'} P(e_h \equiv e') = 0$ for all $(e_h, e'_h) \in \mathcal{L}'$. This trivial solution would lead to the exclusion of most accurate labels, an outcome we aim to avoid. Considering this, we introduce an $l1$ penalty term to penalize the removal of labels, leading to our overall objective:

$$\mathcal{L}^* = \text{argmin}_{\mathcal{L}'\subset\mathcal{L}}\left(\Phi(\mathcal{L}') + \lambda|\mathcal{L} - \mathcal{L}'|\right). \tag{8}$$

https://doi.org/10.52202/079017-0482

Here $\lambda > 0$ is a weight coefficient. Solving the above combinatorial problem is intractable as it requires computing $\Phi(\mathcal{L}')$ for each possible set $\mathcal{L}' \subset \mathcal{L}$, which is NP-hard. Below we propose to search for a local-optimal solution by a greedy algorithm powered by probabilistic reasoning.

Greedy search. The algorithm begins by initializing the alignment probability $P(e \equiv e') = \delta_0$ for every pair (e, e') within the set $\mathcal{L}$, where δ_0 is a constant within the range (0,1). It then iteratively performs a search for an optimal label set $\mathcal{L}'$ through a series of voting steps. Each iteration is comprised of two main steps: probabilistic reasoning and label adjustment.

During the probabilistic reasoning step, the alignment probabilities and subrelation probabilities are updated according to Eq. (5) and Eq. (6), respectively. This update process refines our estimates of label confidence based on the latest information. During the label adjustment step, the label set $\mathcal{L}'$ is updated based on these updated probabilities. Labels are appended to $\mathcal{L}'$ if their updated alignment probabilities exceed δ_0, indicative of high confidence in their alignment, supported by their neighbors:

$$\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(e_h, e'_h) \in \mathcal{L} | P(e_h \equiv e'_h) > \delta_0\}. \tag{9}$$

Conversely, labels demonstrating structural incompatibilities are excluded from $\mathcal{L}'$:

$$\mathcal{L}' \leftarrow \mathcal{L}' \setminus \left\{(e_h, e'_h) \in \mathcal{L} \mid P(e_h \equiv e'_h) < \max\left(\max_{e \in \mathcal{E}} P(e, e'_h), \max_{e' \in \mathcal{E}'} P(e_h, e')\right)\right\}. \tag{10}$$

In this manner, labels are removed if they are incompatible with updated aligned neighbors, ensuring the preservation of only the most confident pairs within $\mathcal{L}'$. To further refine the search process in subsequent iterations, we augment all entity alignment probabilities within $\mathcal{L}'$ to a superior score:

$$P(e \equiv e') \leftarrow \max\left(P(e \equiv e'), \delta_1\right) \quad \text{for each } (e, e') \in \mathcal{L}'. \tag{11}$$

Here $\delta_1 \in (\delta_0, 1)$ serves as a new threshold, elevating the alignment probabilities of confident pairs to foster a more directed and effective search. After n_{lr} iterations, we get a set of confidently selected labels $\mathcal{L}^*$ that have high compatibility. The detailed algorithm is presented in Appendix A.2, and analyses of parameter efficiency and computational efficiency are provided in Appendix A.3.

3.4 Entity alignment

With the refined labels, we train an embedding-based EA model to learn structure-aware representations for each entity. After training, the EA model computes a matching score $m_\theta(e, e')$ for each entity pair (e, e') for evaluation. The selection of the base EA model is flexible, tailored to the task requirements. We chose a recently proposed GCN-based model, Dual-AMN (Mao et al., 2021), for its effectiveness and efficiency.

Feedback from the base EA model is crucial for dynamic update of the active selection policy. To generate effective feedback, we infer high-confidence pairs (e, e') with the trained EA model, by selecting the pairs that both entities rank top for each other. These pairs are injected into the probabilistic reasoning system. Similar to the label refinement process, this system initializes with an alignment probability of δ_0 for these pairs and updates the estimation of alignment and subrelation probabilities using Eq. (5) and Eq. (6). The updated probabilities are used to construct the uncertainty terms (i.e., U_r and U_n) to inform the active selection policy in subsequent iterations, thereby optimizing the budget utility and improving final performance continuously.

4 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our framework. We begin by introducing the experimental settings. Then, we present experiments to answer the following research questions: **RQ1.** How effective is the overall framework? **RQ2.** What is the impact of the choice of LLM on the cost and performance of LLM4EA? **RQ3.** What is the effect of the label refiner? **RQ4.** What is the impact of active selection?

4.1 Experimental setting

Datasets and LLM. In this study, we use the widely-adopted OpenEA dataset (Sun et al., 2020), including two monolingual datasets (D-W-15K and D-Y-15K) and two cross-lingual datasets (EN-DE-15K and EN-FR-15K). OpenEA comes in two versions: "V1" the normal version, and "V2"

the dense version. We employ "V2" in the experiments in the main text. The LLM version in this experiment is GPT-3.5 (gpt-3.5-turbo-0125) and GPT-4 (gpt-4-turbo-2024-04-09). By default, the overall query budget is $\mathcal{B} = 0.1|\mathcal{E}|$.

Baselines. Baseline models include three GCN-based models — GCNAlign (Wang et al., 2018), RDGCN (Wu et al., 2019), Dual-AMN (Mao et al., 2021), and three translation-based models — IMUSE (He et al., 2019), AlignE, BootEA (Sun et al., 2018), Here, BootEA is a variant of AlignE that adopts a bootstrapping strategy, equipped with a label calibration component for improving the accuracy of bootstrapped labels. Baseline models are directly trained on the pseudo-labels generated by the LLM annotator, without label refinement or active selection. Every experiment is repeated three times to report statistics.

Setup of LLM4EA. We employ GPT-3.5 as the default LLM due to its cost efficiency. Other parameters are $n = 3$, n_{lr}, $k = 20$, $\delta_0 = 0.5$, $\delta_1 = 0.9$.

4.2 Results

4.2.1 Comprehensive evaluation of entity alignment performance

Table 1: Evaluation of entity alignment performance, measured by Hit@K for $K \in \{1, 10\}$, and Mean Reciprocal Rank (MRR), presented in %. Experiment statistics are computed over three trials.

	EN-FR-15K			EN-DE-15K			D-W-15K			D-Y-15K		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
Group1. Entity Alignment with GPT-3.5.												
IMUSE	50.0±0.1	72.6±0.8	57.5±0.4	51.6±4.7	75.9±3.9	60.5±4.5	6.0±0.2	14.6±2.5	9.0±1.0	54.4±2.5	78.9±1.1	63.2±2.0
AlignE	6.6±0.3	24.5±0.5	12.6±0.5	6.2±0.3	18.4±1.0	10.4±0.5	8.0±0.9	24.0±2.7	13.3±1.4	50.1±2.0	76.6±1.4	59.2±1.8
BootEA	44.8±1.1	71.9±1.2	54.2±1.2	68.1±0.2	85.4±0.3	74.3±0.2	60.8±0.2	79.3±0.1	67.4±0.2	87.8±0.1	96.7±0.1	91.2±0.1
GCNAlign	17.4±0.3	43.2±0.4	25.9±0.3	22.2±0.2	46.2±1.1	30.3±0.3	16.9±0.1	39.3±0.3	24.3±0.1	45.3±0.4	68.3±0.6	53.3±0.5
RDGCN	69.3±0.3	82.5±0.3	74.3±0.3	73.3±4.3	84.6±2.6	77.4±3.7	79.2±0.7	89.7±0.5	83.2±0.6	82.6±3.7	91.9±1.3	86.1±2.7
Dual-AMN	51.9±0.3	79.6±0.9	61.6±0.5	70.5±0.7	91.1±0.3	78.9±0.6	62.0±0.1	86.8±0.1	71.9±0.1	85.8±0.3	98.4±0.0	91.4±0.1
LLM4EA	**74.2±0.3**	**92.9±0.4**	**81.0±0.3**	**89.1±0.5**	**97.8±0.1**	**92.6±0.3**	**87.5±0.3**	**96.7±0.1**	**90.9±0.2**	**97.7±0.0**	**99.5±0.0**	**98.3±0.0**
Group2. Entity Alignment with GPT-4.												
IMUSE	52.7±0.9	74.9±1.0	59.8±0.9	59.6±2.6	81.8±1.5	67.9±2.1	21.6±6.1	50.0±10.0	31.1±7.4	86.6±0.5	94.2±0.1	89.2±0.4
AlignE	30.8±2.4	69.1±2.5	43.1±2.5	46.4±5.2	76.5±3.8	56.6±4.8	36.1±3.7	67.8±3.6	46.7±3.7	86.4±0.9	97.0±0.3	90.2±0.6
BootEA	58.2±0.3	83.7±0.3	67.0±0.3	80.5±0.4	92.6±0.2	84.8±0.3	71.6±0.2	88.3±0.2	77.6±0.2	95.0±0.1	98.6±0.0	96.3±0.1
GCNAlign	30.6±0.0	65.3±0.3	42.1±0.2	41.9±0.4	68.6±0.5	51.2±0.4	31.3±0.3	61.6±0.1	41.4±0.2	82.6±0.2	94.9±0.2	87.2±0.1
RDGCN	72.1±0.2	84.5±0.1	76.7±0.2	74.1±1.1	85.1±0.7	78.0±1.0	82.5±1.1	91.4±0.7	85.9±1.0	85.4±0.9	93.2±0.4	88.3±0.8
Dual-AMN	76.7±0.1	94.9±0.3	83.6±0.2	90.7±0.1	97.9±0.2	93.6±0.1	81.5±0.1	94.9±0.2	86.7±0.1	97.5±0.0	99.3±0.1	98.1±0.0
LLM4EA	**80.2±0.3**	**96.0±0.2**	**86.0±0.2**	**93.1±0.5**	**98.7±0.2**	**95.3±0.3**	**89.8±0.3**	**97.9±0.2**	**92.9±0.3**	**97.9±0.1**	**99.6±0.0**	**98.5±0.1**

To answer **RQ1** and **RQ2**, we conducted two groups of experiments on OpenEA datasets, using GPT-3.5 and GPT-4 as the annotator, respectively. Results are presented in Table 1. We also investigated the performance-cost comparison between the GPT-3.5 annotator and the GPT-4 annotator, illustrated in Figure 2. To control the randomness introduced by the LLMs, each experiment was repeated three times to report mean and standard deviation. These results lead to several key observations:

First, LLM4EA surpasses all baseline EA models, which are directly trained on the pseudo-labels, by a large margin. This can be attributed to 1) our label refiner's capability in filtering out false labels, reducing noise during training and enabling more accurate optimization towards the ground true objective; 2) our active selection component's ability to smartly identify important entities to annotate, which takes full advantage of the fixed query budget.

Second, using the GPT-4 results in higher performance than using the GPT-3.5 as the annotator. This observation conforms to the fact that GPT-4 is a more advanced LLM with higher reasoning capacity and stronger semantic analysis, resulting in more precise annotation results and higher recall, thus providing more labels of high quality. We also observe that translation-based models (e.g., AlignE) are sensitive to noisy labels under GPT-3.5, while state-of-the-art GCN-based models (e.g., RDGCN and Dual-AMN) are more robust. BootEA also demonstrates superior performance and robustness, attributed to its bootstrapping technique and enhanced by its capability in calibrating bootstrapped labels. However, its label calibration is only applied to the bootstrapped labels, so it still suffers from the false training labels. Our proposed LLM4EA, on the other hand, refines the label accuracy before training the EA model, thus ensuring more accurate training.

https://doi.org/10.52202/079017-0482

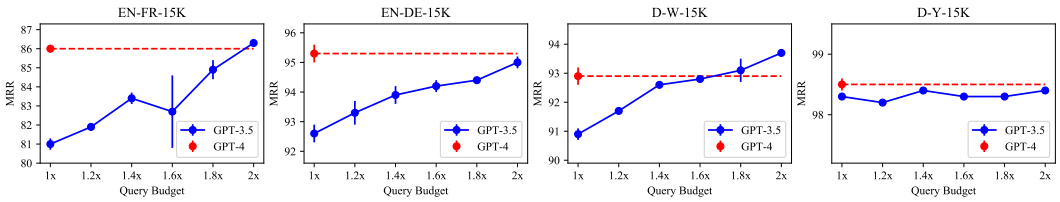

Figure 2: Performance-cost comparison between GPT-3.5 and GPT-4 as the annotator, evaluated by MRR. We increase the budget for GPT-3.5 to evaluate its performance. $[n\times]$ denotes using $n\times$ of the default query budget. Each experiment is repeated three times to show mean and standard deviation.

Finally, LLM4EA is noise adaptive, enabling cost-efficient entity alignment. To further investigate the effect of the choice of LLM, we examined the performance-cost comparison between GPT-3.5 and GPT-4 as the annotator. We illustrate MRR in Figure 2 (detailed results are available in Appendix B.3). The results show that, by increasing the query budgets (measured by the number of tokens) for GPT-3.5, the performance gradually increases. When the budget is $2\times$ that of GPT-4, the performance is comparable to or exceeds the performance of using GPT-4 as the annotator. According to the pricing scheme of OpenAI, the input/output cost for 1 million tokens for GPT-3.5 and GPT-4 is \$0.50/\$1.5 and \$10/\$30, respectively. This means that our noise-adaptive framework enables cost-efficient entity alignment with less advanced LLMs at $10\times$ less actual cost than using more advanced LLMs, simply by increasing the token budget for the less advanced LLMs.

4.2.2 Effect of the label refiner

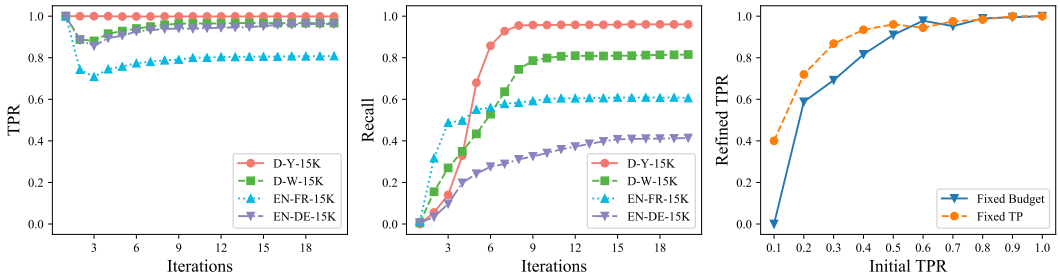

Figure 3: Analysis of the Label Refinement. We illustrate the evolution of the true positive rate (TPR) (left) and recall (middle) for refined labels across four datasets. Furthermore, we assess the robustness of the label refinement process by examining the TPR of refined labels against varying initial TPRs within the D-W-15K dataset (right), with initial pseudo-labels synthesized at different TPR levels.

To answer **RQ3**, we first analyze the evolution of the True Positive Rate (TPR) and the recall rate of the refined labels. Specifically, at each label refinement iteration, the TPR is calculated as $\frac{|\mathcal{A}\cap\mathcal{L}'|}{|\mathcal{L}'|}$, and the recall is calculated as $\frac{|\mathcal{A}\cap\mathcal{L}'|}{|\mathcal{A}\cap\mathcal{L}|}$. The left and middle subfigures of Figure 3 demonstrate how **our label refiner progressively discovers accurate labels and optimizes the TPR.** Initially, the TPR of the refined label set is high (approximately 1.0), then it decreases by a certain percentage, and eventually increases again to a high TPR. We attribute this pattern to: 1) the most confident labels being discovered in the earliest iterations, which are obvious alignments with many connected alignments; 2) as the algorithm progresses, some false pseudo-labels being erroneously added to the label set $\mathcal{L}'$; 3) as the label refinement continues, $\mathcal{L}'$ is adjusted and the false pseudo-labels are replaced with the correct labels inferred by the updated probability as in Eq. (10).

Furthermore, we assess the robustness of our label refiner, as depicted in the right subfigure of Figure 3. We synthesize noisy labels and evaluate the output TPR in relation to varying input TPR levels, using two experimental schemes: *fixed budget*, where the budget remains constant at $0.1|\mathcal{E}|$ while the TPR changes, and *fixed TP*, where the number of true positives is fixed but the TPR and corresponding budgets are adjusted. The results demonstrate that **the label refiner consistently elevates the TPR to over** 0.9**, even when the initial TPR is around** 0.5**, showcasing its high**

https://doi.org/10.52202/079017-0482

robustness to noisy pseudo-labels. This result also reveals why our framework demonstrates robust performance with the less advanced GPT-3.5 annotator.

4.2.3 Ablation study

Table 2: Ablation study overview. The table presents the performance of the LLM4EA (Ours) with various modifications. **Group 1**: removing the label refiner (w/o LR) and the active selection component (w/o Act); **Group 2**: replacing the active selection technique with relational uncertainty (-ru), neighbor uncertainty (-nu), degree (-degree), and functionality sum (-funcSum).

	EN-FR-15K			EN-DE-15K			D-W-15K			D-Y-15K		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
Ours	74.2±0.3	92.9±0.4	81.0±0.3	**89.1±0.5**	**97.8±0.1**	**92.6±0.3**	87.5±0.3	96.7±0.1	90.9±0.2	**97.7±0.0**	**99.5±0.0**	**98.3±0.0**
w/o LR	51.6±1.0	80.2±0.7	61.9±0.8	74.4±1.7	94.2±0.6	82.6±1.3	39.2±1.4	75.7±0.7	52.9±1.2	85.3±1.0	99.2±0.1	91.5±0.5
w/o Act	68.1±2.1	88.4±1.7	75.4±2.0	78.4±0.8	93.9±0.3	84.6±0.6	82.8±0.7	92.5±0.6	86.3±0.6	97.5±0.1	99.2±0.3	98.1±0.1
Ours-ru	70.8±1.0	91.2±0.4	78.2±0.8	83.9±0.3	97.7±0.2	89.6±0.2	**88.7±0.6**	**97.4±0.3**	**92.1±0.5**	**97.7±0.0**	99.4±0.1	**98.3±0.0**
Ours-nu	**74.5±0.7**	**93.1±0.5**	**81.2±0.6**	88.8±0.2	96.7±0.3	91.8±0.3	85.1±0.5	95.2±0.5	88.9±0.5	97.6±0.1	99.4±0.0	98.2±0.0
Ours-degree	73.6±2.6	92.5±0.8	80.4±2.0	88.4±0.1	96.6±0.2	91.5±0.2	80.1±3.7	90.9±2.2	84.0±3.2	97.2±0.2	99.0±0.1	97.9±0.1
Ours-funcSum	59.5±0.6	78.8±0.6	66.3±0.6	81.2±0.5	96.0±0.3	87.1±0.4	83.9±0.9	93.1±1.1	87.3±1.0	97.5±0.1	99.4±0.1	98.1±0.1

Ablation studies detailed in Table 2 answer **RQ4** and reveal several key insights: **1) Necessity of the Label Refiner for Effective Active Selection:** The performance of "w/o LR", which lacks a label refiner, is inferior not only to other model variants but also to the base model, Dual-AMN. This underscores that active selection depends crucially on reliable feedback, which is compromised when the label refiner is absent; **2) Contribution of Relation and Neighbor Uncertainty in Active Selection:** Incorporating both relation and neighbor uncertainties significantly enhances the utility of the budget. Methods like "Ours-degree" and "Ours-funcSum" focus only on their connections to neighbors and ignore the uncertainty of neighbors. In contrast, "Ours-ru" and "Ours-nu", which take these uncertainties into account, exhibit superior performance. This underscores the importance of considering neighbor uncertainty for effective active selection; **3) Robust Active Selection through Combined Metrics:** Our active selection approach integrates both relation uncertainty and neighbor uncertainty to enable robust active selection. By employing rank aggregation, it prioritizes entities that are deemed significant by both metrics, ensuring a more effective and nuanced selection process.

4.2.4 Pareto frontier of runtime overhead against performance.

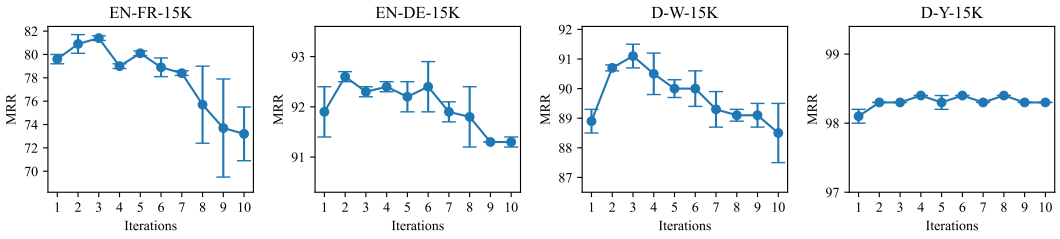

Figure 4: Performance of entity alignment across four datasets with varying active sampling iterations, under a fixed query budget.

The runtime overhead is directly proportional to the number of active selection iterations, n, since each iteration involves a subsequent label refinement process. To explore the runtime-performance trade-off in the LLM4EA approach, we examine the Pareto frontier of runtime versus performance. We conduct entity alignment experiments with a fixed query budget, varying the number of active selection iterations. The results of these experiments are illustrated in Figure 4.

The results indicate that performance initially increases as the number of iterations rises from 1 to around 3, but further increases beyond this point lead to a decline. This pattern can be attributed to two main factors: (1) More iterations allow for extensive learning from feedback during the active selection phase. (2) However, when iterations are excessively high, the number of generated pseudo-labels per iteration becomes small, leading to isolated pseudo-labels that undermine the label refinement process. These findings suggest that **an optimal balance between runtime efficiency and performance can be achieved without excessive trade-offs**, indicating a specific threshold for iterations beyond which no further performance gains are observed.

https://doi.org/10.52202/079017-0482

5 Related work

LLMs for Entity Alignment. Recent approaches have sought to leverage LLMs for entity alignment in knowledge graphs, primarily focusing on integrating structural and semantic information for improved alignment performance. BERT-INT (Tang et al., 2020) fine-tunes a pretrained BERT model to capture interactions and attribute information between entities. Similarly, SDEA (Zhong et al., 2022) employs a pretrained BERT to encode attribute data, while integrating neighbor information via a GRU to enhance structural representation. TEA (Zhao et al., 2023b) reconceptualizes entity alignment as a bidirectional textual entailment task, utilizing pretrained language models to estimate entailment probabilities between unified textual sequences representing entities. A novel approach, ChatEA (Jiang et al., 2024), introduces a KG-code translation module that converts knowledge graph structures into a format comprehensible to LLMs, enabling these models to apply their extensive background knowledge to boost the accuracy of entity alignment. Notably, these models primarily focus on fine-tuning pretrained language models using a set of training labels and do not exploit the zero-shot capabilities of LLMs. In contrast, our proposed model, LLM4EA, leverages the zero-shot potential of LLMs, enabling it to generalize to new datasets without the need for labeled data.

Probabilistic Reasoning. In the literature on knowledge graph reasoning, probabilistic reasoning has been effectively applied to infer new soft labels and their associated probabilities from existing labels. It has been utilized in domains such as knowledge graph completion (Qu & Tang, 2019; Zhang et al., 2020; Fang et al., 2023; Chen et al., 2024b) and entity alignment (Suchanek et al., 2012; Qi et al., 2021; Liu et al., 2022; Chen et al., 2024c), where it naturally represents complex relational patterns with simple rules and performs precise inferences. In this work, however, due to the potential inaccuracies in the pseudo-labels generated by LLMs, the newly inferred alignments may be incorrect. Consequently, we opt not to employ probabilistic reasoning directly for the entity alignment task. Instead, we emphasize its use primarily for filtering false pseudo-labels that demonstrate structural incompatibilities within our framework. For completeness, we include the results of comparison with a probabilistic reasoning model – PARIS (Suchanek et al., 2012) in Appendix B.4.

6 Limitations

Firstly, during active selection, we distribute the query budget evenly for each selection rather than dynamically customizing the budget allocation for each iteration. This allocation approach could be improved by developing a more adaptive budget allocation strategy. Secondly, the framework currently does not provide direct support for temporal KGs. Although the probabilistic reasoning and active selection components inherently support entity alignment on evolving KGs, the base EA model is transductive. This necessitates retraining the model whenever new entities and relation triples are introduced into the KGs. However, this can be complemented by the research line of inductive entity alignment, such as path-based embedding models or logic-based models, which can generalize to previously unseen entities without the need for retraining.

7 Conclusions

In this paper, we address the challenge of automating entity alignment with Large Language Models (LLMs) under budget constraints and noisy annotations. We introduce LLM4EA, a framework that maximizes the utility of a fixed query budget using active sampling and mitigates erroneous labels with a label refiner employing probabilistic reasoning. Empirical results show that LLM4EA's noise-adaptive capabilities reduce costs without sacrificing performance, achieving comparable or superior results with less advanced LLMs at up to 10 times lower cost. This highlights the potential of LLM-based models in merging cross-domain and cross-lingual KGs. Future work will explore incorporating real-time learning capabilities to dynamically adjust to evolving knowledge bases.

Acknowledgement

The work was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15200023).

References

Rose Catherine and William Cohen. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM conference on recommender systems*, pp. 325–332, 2016.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*, 2023.

Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. Macro graph neural networks for online billion-scale recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pp. 3598–3608, 2024a.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1511–1517, 2017.

Shengyuan Chen, Yunfeng Cai, Huang Fang, Xiao Huang, and Mingming Sun. Differentiable neuro-symbolic reasoning on large-scale knowledge graphs. *Advances in Neural Information Processing Systems*, 36, 2024b.

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Jiannong Cao, and Xiao Huang. Neuro-symbolic entity alignment via variational inference. *arXiv preprint arXiv:2410.04153*, 2024c.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (LLMs). In *The Twelfth International Conference on Learning Representations*, 2024d.

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. Hierarchy-aware multi-hop question answering over knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pp. 2519–2527, 2023.

Junnan Dong, Zijin Hong, Yuanchen Bei, Feiran Huang, Xinrun Wang, and Xiao Huang. Clr-bench: Evaluating large language models in college-level reasoning, 2024a. URL https://arxiv.org/abs/2410.17558.

Junnan Dong, Qinggang Zhang, Chuang Zhou, Hao Chen, Daochen Zha, and Xiao Huang. Cost-efficient knowledge-based question answering with large language models. *arXiv preprint arXiv:2405.17337*, 2024b.

Junnan Dong, Qinggang Zhang, Huachi Zhou, Daochen Zha, Pai Zheng, and Xiao Huang. Modality-aware integration with large language models for knowledge-based visual question answering, 2024c.

Huang Fang, Yang Liu, Yunfeng Cai, and Mingming Sun. Mln4kb: an efficient markov logic network engine for large-scale knowledge bases and structured logic rules. In *Proceedings of the ACM Web Conference 2023*, pp. 2423–2432, 2023.

Fuzhen He, Zhixu Li, Yang Qiang, An Liu, Guanfeng Liu, Pengpeng Zhao, Lei Zhao, Min Zhang, and Zhigang Chen. Unsupervised entity alignment using attribute triples and relation triples. In *Database Systems for Advanced Applications: 24th International Conference, DASFAA 2019, Chiang Mai, Thailand, April 22–25, 2019, Proceedings, Part I 24*, pp. 367–382, 2019.

Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. Aligning distillation for cold-start item recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1147–1157, 2023.

Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. Unlocking the power of large language models for entity alignment. *arXiv preprint arXiv:2402.15048*, 2024.

Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10*, pp. 273–288, 2011.

https://doi.org/10.52202/079017-0482

Zhixun Li, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, Liang Wang, et al. Gslb: the graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.

Bing Liu, Harrisen Scells, Wen Hua, Guido Zuccon, Genghong Zhao, and Xia Zhang. Guiding neural entity alignment with compatibility. *arXiv preprint arXiv:2211.15833*, 2022.

Yang Liu, Huang Fang, Yunfeng Cai, and Mingming Sun. Mquine: a cure for" z-paradox"in knowledge graph embedding models. *arXiv preprint arXiv:2402.03583*, 2024a.

Yang Liu, Chuan Zhou, Peng Zhang, Yanan Cao, Yongchao Liu, Zhao Li, and Hongyang Chen. Cl4kge: A curriculum learning method for knowledge graph embedding. *arXiv preprint arXiv:2408.14840*, 2024b.

Zirui Liu, Chen Shengyuan, Kaixiong Zhou, Daochen Zha, Xiao Huang, and Xia Hu. Rsc: accelerate graph neural networks training via randomized sparse computations. In *International Conference on Machine Learning*, pp. 21951–21968. PMLR, 2023.

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

Zhiyuan Qi, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Yuejia Xiang, Ningyu Zhang, and Yefeng Zheng. Unsupervised knowledge graph alignment by probabilistic reasoning and semantic embedding. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2019–2025, 2021.

Meng Qu and Jian Tang. Probabilistic logic neural networks for reasoning. *Advances in neural information processing systems*, 32, 2019.

Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. In *Proceedings of the 38th International Conference on Very Large Databases*, pp. 157–168, 2012.

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

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(12):2326–2340, 2020.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998, 2008.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. Bert-int:a bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3174–3180, 2020.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19206–19214, 2024.

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

Gerhard Weikum and Martin Theobald. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 65–76, 2010.

Xuansheng Wu, Huachi Zhou, Wenlin Yao, Xiao Huang, and Ninghao Liu. Towards personalized cold-start recommendation with prompts. *arXiv preprint arXiv:2306.17256*, 2023.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

Qinggang Zhang, Junnan Dong, Hao Chen, Wentao Li, Feiran Huang, and Xiao Huang. Structure guided large language model for sql generation. *arXiv preprint arXiv:2402.13284*, 2024a.

Qinggang Zhang, Junnan Dong, Hao Chen, Daochen Zha, Zailiang Yu, and Xiao Huang. Knowgpt: Knowledge injection for large language models, 2024b.

Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. In *International Conference on Learning Representations*, 2020.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023a.

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

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

Huachi Zhou, Jiaqi Fan, Xiao Huang, Ka Ho Li, Zhenyu Tang, and Dahai Yu. Multi-interest refinement by collaborative attributes modeling for click-through rate prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4732–4736, 2022.

 https://doi.org/10.52202/079017-0482

A Notations and algorithms

A.1 Notations

Table 3: Notations.

Notation	Description
$\mathcal{G}, \mathcal{G}'$	The source and target knowledge graphs, respectively
$\mathcal{E}, \mathcal{E}'$	The sets of entities in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{R}, \mathcal{R}'$	The sets of relations in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{T}, \mathcal{T}'$	The sets of relational triplets in $\mathcal{G}$ and $\mathcal{G}'$, respectively
$\mathcal{A}$	The ground truth set of aligned entity pairs
$\mathcal{B}$	The query budget for the LLM
$\mathcal{F}(r)$	The functionality of relation r
$P(e \equiv e')$	The alignment probability of entity pair (e, e')
$P(e) = \max_{e' \in \mathcal{E}'} P(e \equiv e')$	The alignment probability of the top-match entity for e
$\mathcal{L}, \mathcal{L}^*$	The annotated pseudo-label set and the refined pseudo-label set
$\Phi(\mathcal{L})$	The overall incompatibility of the pseudo-label set $\mathcal{L}$
n	The number of iterations of active selection
n_{lr}	The number of iterations of label refinement

A.2 Pseudo-code of the greedy algorithm

Below we present the pseudo-code of the greedy algorithm, that incorporates probabilistic reasoning to refine the label set.

Algorithm 1 The greedy label refinement algorithm

Inputs: The pseudo-label set $\mathcal{L}$
Parameters: The intialization probability $\delta_0 \in (0, 1)$, the threshold $\delta_1 \in (\delta_0, 1)$, probabilistic reasoning iterations n_{lr}
Outputs: The refined pseudo-label set $\mathcal{L}^* \subset \mathcal{L}$
$\mathcal{L}' \leftarrow \varnothing$.
$\forall e \in \mathcal{E}, \forall e' \in \mathcal{E}', P(e \equiv e') \leftarrow 0$
$\forall (e, e') \in \mathcal{L}, P(e \equiv e') \leftarrow \delta_0$
$i \leftarrow 0$
while $i < n_{lr}$ **do**
$\forall e_h \in \mathcal{E}, \forall e'_h \in \mathcal{E}', P(e_h \equiv e'_h) \leftarrow 1 - \prod_{\substack{(e_h, r, e_t) \in \mathcal{T}, \\ (e'_h, r', e'_t) \in \mathcal{T}'}} \left(1 - \mathcal{F}^{-1}(r) P(r \subseteq r') P(e_t \equiv e'_t)\right) \times \left(1 - \mathcal{F}^{-1}(r') P(r' \subseteq r) P(e_t \equiv e'_t)\right)$. /* Update entity alignment probabilities.*/
$\forall r \in \mathcal{R}, \forall r' \in \mathcal{R}', P(r \subseteq r') \leftarrow \frac{\sum\left(1 - \prod_{(e'_h, r', e'_t) \in \mathcal{T}'} \left(1 - P(e'_h \equiv e_h) P(e'_t \equiv e_t)\right)\right)}{\sum\left(1 - \prod_{e'_h, e'_t \in \mathcal{E}'} \left(1 - P(e'_h \equiv e_h) P(e'_t \equiv e_t)\right)\right)}$ /* Update subrelation probabilities. */
$\mathcal{L}' \leftarrow \mathcal{L}' \cup \{(e_h, e'_h) \in \mathcal{L} | P(e_h \equiv e'_h) > \delta_0\}$ /* Label adjustment, add confident pairs to the label set.*/
$\mathcal{L}' \leftarrow \mathcal{L}' \setminus \{(e_h, e'_h) \in \mathcal{L} \mid P(e_h \equiv e'_h) < \max\left(\max_{e \in \mathcal{E}} P(e, e'_h), \max_{e' \in \mathcal{E}'} P(e_h, e')\right)\}$ /* Label adjustment, remove less confident pairs from the label set. */
$P(e \equiv e') \leftarrow \max\left(P(e \equiv e'), \delta_1\right)$ for each $(e, e') \in \mathcal{L}'$ /* Elevate alignment probability of confident pairs. */
end while
$\mathcal{L}^* \leftarrow \mathcal{L}' \cup \{(e, e') | P(e \equiv e') > \delta_1\}$ /* Augment the refined label set with confident pairs.*/
Return $\mathcal{L}^*$

A.3 Efficient implementation of label refiner

Parameter-efficient probabilistic reasoning. The total number of alignment probabilities for all entity pairs is $|\mathcal{E}||\mathcal{E}'|$, resulting in a large parameter size when the KGs involved are extensive. We enhance memory efficiency by adopting a lazy inference strategy in probabilistic reasoning. This

strategy involves only saving the alignment probabilities of the most probable alignments:

$$\left\{P(e_h, e'_h), |, e_h \in \mathcal{E}, e'_h \in \mathcal{E}', P(e_h \equiv e'_h) = \max\left(\max_{e\in\mathcal{E}} P(e, e'_h), \max_{e'\in\mathcal{E}'} P(e_h, e')\right)\right\}, \quad (12)$$

Probabilities of other entity pairs can be inferred from these saved alignment probabilities using Eq. (5) when necessary. In this way, parameter complexity is reduced to $O(\max(|\mathcal{E}| + |\mathcal{E}'|))$.

Computation efficiency of probabilistic reasoning. Probabilistic reasoning is executed iteratively, with each iteration updating the alignment probabilities of all entities and relations following Eq. (5) and Eq. (6). We detail the analysis of these two update phases in the following separately.

Since we adopt a lazy inference strategy, the update of entity alignment probabilities involves updating the set of most probable alignments and associated probabilities as shown in Eq. (12). This update requires the estimation of all $P(e, e'_h), e \in \mathcal{E}$ and all $P(e_h, e'), e' \in \mathcal{E}'$, for each current pair (e_h, e'_h) to determine if this pair needs an update. Consequently, the computational complexity of this process is proportional to the size of this set, which is $O(|\mathcal{E}|)$. The estimated scores $P(e, e'_h), e \in \mathcal{E}$ and $P(e_h, e'), e' \in \mathcal{E}'$ can be precomputed in advance and reused for all pairs (e_h, e'_h), leading to a computation complexity of $O(|\mathcal{E}||\mathcal{E}'|)$. Thus, the overall computational complexity for updating entity alignment probabilities is $O(|\mathcal{E}||\mathcal{E}'| + |\mathcal{E}|) = O(|\mathcal{E}||\mathcal{E}'|)$.

The update process for sub-relation probabilities involves updating all $P(r \subset r')$ for $r \in \mathcal{R}$ and $r' \in \mathcal{R}'$, resulting in a complexity of $O(|\mathcal{R}||\mathcal{R}'|)$. The estimation of Eq. (6) utilizes the probabilities of the most probable alignments from Eq. (12). Notably, most relation pairs (r, r') do not have aligned head entities (e_h, e'_h) or aligned tail entities (e_t, e'_t), thus most relation pairs can be excluded for efficient computation by exploiting this sparsity heuristic, reducing the computations by orders.

It is worth noting that these computations can be further accelerated through parallelization, as their execution solely depends on the results from the previous iteration.

B Experimental details

B.1 Hardware and software configurations

Our experiments were conducted on a server equipped with six NVIDIA GeForce RTX 3090 GPUs, 48 Intel(R) Xeon(R) Silver 4214R CPUs, and 376GB of host memory. The models were implemented using TensorFlow, NumPy, and SciPy. It was observed that the software version significantly affects hardware-software compatibility. Specifically, the original implementation of Dual-AMN was based on TensorFlow 1.14.0, which is not compatible with newer GPUs such as the NVIDIA GeForce RTX 3090. Therefore, we updated the code to be compatible with TensorFlow 2.7.0, enabling the model to leverage GPU acceleration effectively. The details of the software packages used in our experiments are listed in Table 4.

Table 4: Package configurations of our experiments.

Package	tqdm	numpy	scipy	tensorflow	keras	openai
Version	4.66.2	1.24.4	1.10.1	2.7.0	2.7.0	1.30.1

B.2 Dataset statistics and preprocessing

In our experiments, we utilized the OpenEA dataset version 1.1 (V2), specifically the 15K set. The statistics are detailed in Table 5. It's important to highlight that the destination dataset for D-W-15K, originating from Wikidata, contains only entity IDs, lacking explicit names. These IDs, devoid of semantic content, are not inherently meaningful to a language model. To rectify this, we processed the dataset using the 'wikidatawiki-20160801-abstract.xml' dump file from Wikidata. This file provided the raw data necessary for constructing the D-W-15K dataset, enabling us to extract meaningful entity names.

We shown the quantity of entities with names (after name extraction) for each KG in the 'Named entities' column in Table 5.

https://doi.org/10.52202/079017-0482

Table 5: Data statistics of used OpenEA dataset.

Datasets	KG	Relations	Relation triplets	Attributes	Attribute triples	Named Entities	Targets in top-k
EN-FR	English (EN)	193	96,318	189	66,899	15,000	13,550
	French (FR)	166	80,112	221	68,779	15,000	
EN-DE	English (EN)	169	84,867	171	81,988	15,000	13,330
	German (DE)	96	92,632	116	186,335	15,000	
D-W	DBpedia (DB)	167	73,983	175	66,813	15,000	12,910
	Wikidata (WD)	121	83,365	457	175,686	13,458	
D-Y	DBpedia (DB)	72	68,063	90	65,100	15,000	15,000
	Yago (YG)	21	60,970	20	131,151	15,000	

In the counterpart filtering phase, we selected the top-k (where $k = 20$ for our experiments) most similar candidates. The 'Target in top-k' column of Table 5 shows the number of target entities included in this selection.

B.3 Performance-cost comparison between GPT-3.5 and GPT-4

Table 6: Performance-cost comparison between GPT-3.5 and GPT-4 as annotator. We increase the budget for GPT-3.5 to evaluate its performance. [$n\times$] denotes using $n\times$ of the default query budget.

	EN-FR-15K			**EN-DE-15K**			**D-W-15K**			**D-Y-15K**		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
GPT-3.5	74.2±0.3	92.9±0.4	81.0±0.3	89.1±0.5	97.8±0.1	92.6±0.3	87.5±0.3	96.7±0.1	90.9±0.2	97.7±0.0	99.5±0.0	98.3±0.0
GPT-3.5 (1.2×)	75.4±0.4	93.2±0.4	81.9±0.2	90.2±0.6	98.0±0.0	93.3±0.4	88.4±0.1	97.1±0.2	91.7±0.1	97.6±0.0	99.3±0.1	98.2±0.0
GPT-3.5 (1.4×)	77.2±0.2	94.5±0.5	83.4±0.3	91.0±0.4	98.1±0.1	93.9±0.3	89.2±0.2	97.9±0.0	92.6±0.1	97.7±0.0	99.5±0.0	98.4±0.0
GPT-3.5 (1.6×)	76.3±2.4	94.1±0.7	82.7±1.9	91.4±0.2	98.4±0.0	94.2±0.2	89.6±0.0	97.9±0.2	92.8±0.1	97.7±0.0	99.4±0.0	98.3±0.0
GPT-3.5 (1.8×)	78.8±0.5	95.2±0.4	84.9±0.5	91.9±0.1	98.1±0.0	94.4±0.1	90.1±0.5	97.9±0.2	93.1±0.4	97.7±0.0	99.5±0.1	98.3±0.0
GPT-3.5 (2×)	**80.6±0.2**	95.9±0.1	**86.3±0.1**	92.7±0.2	98.5±0.1	95.0±0.2	**90.7±0.2**	**98.5±0.1**	**93.7±0.1**	97.8±0.0	99.5±0.0	98.4±0.0
GPT-4	80.2±0.3	**96.0±0.2**	86.0±0.2	**93.1±0.5**	**98.7±0.2**	**95.3±0.3**	89.8±0.3	97.9±0.2	92.9±0.3	**97.9±0.1**	**99.6±0.0**	**98.5±0.1**

B.4 Performance comparison against rule-based models

Table 7: Performance comparison against rule-based models, evaluated by precision, recall, and f1-score in %.

	EN-FR-15K			**EN-DE-15K**			**D-W-15K**			**D-Y-15K**		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Emb-Match	**88.9**	**73.6**	**80.5**	89.7	69.5	78.3	91.8	62.4	74.3	**100**	**100**	**100**
Str-Match	84.8	69.8	76.6	**92.3**	71.4	80.5	**96.2**	57.9	72.3	76.9	**100**	86.9
PARIS	58.3±0.5	26.5±0.3	36.5±0.4	90.8±0.3	50.7±0.3	65.0±0.3	92.4±0.4	70.2±0.2	79.8±0.2	99.1±0.1	96.7±0.1	97.9±0.1
LLM4EA	68.6±0.3	53.1±0.2	59.8±0.2	90.5±0.3	**82.4±0.4**	**86.2±0.4**	90.7±0.4	**81.6±0.5**	**85.9±0.5**	98.9±0.0	97.6±0.1	98.3±0.0

In this section, we compare the LLM4EA model with several rule-based models, including two lexical matching-based approaches: Emb-Match and Str-Match. Emb-Match uses cosine similarities between word embeddings to identify aligned pairs, employing a pretrained language model. Str-Match utilizes the Levenshtein Distance to calculate similarity scores. Additionally, the probabilistic reasoning model PARIS performs entity alignment by relying on probabilistic methods.

For the lexical matching-based models, we compute the similarity and evaluate the confident entity pairs, specifically targeting those whose normalized similarity scores exceed 0.8. The pretrained language model used for Emb-Match is bert-base-uncased for its ability to process unseen words as a subword model. To reduce false positives, we implementing a filtering process that only considers 1-1 matching for embedding-based matching. For PARIS, we assess all inferred aligned pairs by setting the threshold to zero. The entity alignment (EA) model of LLM4EA generates a ranked score list, which is not directly comparable with these rule-based models. To facilitate comparison, we use the trained EA model to generate confident pairs, ensuring that each entity is the top-ranked candidate for its counterpart. These pairs are then processed through our label refiner, and the refined pairs are evaluated. Experiments for PARIS and LLM4EA are repeated three times to ensure statistical reliability; for Emb-Match and Str-Match, experiments are performed once as these algorithms are deterministic. The results are presented in Table 7.

The findings highlight several key observations: 1) Lexical matching-based methods show a degree of alignment ability by leveraging name similarity, particularly on the D-Y-15K dataset, where many aligned entities have identical names. 2) These methods, however, encounter scalability challenges due to their reliance on name similarity. As the size of the KG grows, the number of similar entities increases, making it difficult to maintain precision. This is evident from our experiments on EN-FR-100K and EN-DE-100K, where precision dropped to 49% and recall to 65%, a significant decrease compared to the 15K-size datasets, supporting our analysis. 3) PARIS achieves precise results by handling noisy data through probabilistic reasoning, although its recall is lower than LLM4EA's. 4) LLM4EA, with its active selection and label refinement techniques, consistently delivers robust and accurate performance across all datasets.

https://doi.org/10.52202/079017-0482

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly made the main claims in both the abstract and introduction. Specifically, we have summarized and itemized the contributions at the end of the introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see page 9, Section 6 for the **Limitations** section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the code for the framework, accessible via this URL: https://github.com/chensyCN/llm4ea_official.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

https://doi.org/10.52202/079017-0482

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and datasets are accessible through this anonymous URL: https://anonymous.4open.science/r/llm4ea_neurips2024-E763/. In this repository, we included detailed instructions for running the code in the readme.md file.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We present the experimental setting details at the beginning of the experiment section. We also provide the dataset statistics and the preprocessing details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: To control for the randomness introduced by the Large Language Models, we repeat each experiment three times and report the mean and standard deviation in tables such as Table 1, Table 2, Table 6, and Table 7. For visualized figures such as Figure 2 and Figure 4, we show the error bars representing the standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the computation resources required to reproduce our experiment in Appendix B.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We checked and ensured that our paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper focuses on technical improvements in entity alignment using large language models and does not introduce direct societal impacts.

    Guidelines:

https://doi.org/10.52202/079017-0482

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks, as the work involves only querying the API of an LLM to get pseudo-labels, without releasing a generative model or datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers of the datasets and EA models used in our experiments, ensuring proper credit and respect for their licenses and terms of use. For the open-source code and datasets, we have also stated the licenses they use in the readme.txt file in our code repository.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code of the proposed framework is released and accessible via this anonymouse URL: `https://anonymous.4open.science/r/llm4ea_neurips2024-E763/`. It contains a readme.md file and a GPLv3 license.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

https://doi.org/10.52202/079017-0482

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.