Advancing Cross-Lingual Entity Alignment with Large Language Models: Tailored Sample Segmentation and Zero-Shot Prompts

Linyan Yang, Jingwei Cheng[*], Fu Zhang
School of Computer Science and Engineering, Northeastern University, China
yanglinyanly@163.com, {chengjingwei, zhangfu}@mail.neu.edu.cn

Abstract

In recent years, the advent of large language models (LLMs) like GPT and Llama has significantly influenced numerous domains, particularly in advancing natural language processing (NLP) capabilities. LLMs have shown remarkable performance in NLP tasks such as relation extraction (RE) and knowledge graph completion (KGC), enhancing activities related to knowledge graphs. As a result, there is a growing interest in integrating LLMs into cross-lingual entity alignment (EA) task, which aims to identify equivalent entities across various knowledge graphs, thereby improving the performance of current baselines. However, employing LLMs for entity alignment poses challenges in efficiently handling large-scale data, generating suitable data samples, and adapting prompts for the EA task. To tackle these challenges, we propose Seg-Align, an innovative framework that integrating distance feature extraction, sample **Seg**mentation, and zero-shot prompts. Through extensive experiments on two widely used cross-lingual benchmark datasets, we have not only demonstrated the effectiveness of our proposed sample segmentation algorithm but also highlighted the state-of-the-art performance of Seg-Align. Code is available at `https://github.com/yangxiaoxiaoly/Seg-Align`.

1 Introduction

Knowledge Graphs (KGs) depict entities and their relationships, serving as foundational elements for applications like semantic search (Zhang et al., 2021), question-answering (Kwiatkowski et al., 2019), and recommender systems (Zangerle and Bauer, 2022). However, KGs often suffer from heterogeneity and redundancy due to their construction by various organizations with specific needs. Knowledge fusion (Dong et al., 2014) aims to align and merge this heterogeneous information, forming unified identifiers and relationships. Entity Alignment (EA) is crucial in this process, focusing on discovering equivalent entities across various KGs (Sun et al., 2020b).

Knowledge representation learning-based entity alignment methods have emerged as the primary technique for addressing the EA task, yielding promising results. These SLM based methods[1] often use translation-based models or Graph Neural Networks (GNNs)/ Graph Convolutional Networks (GCNs) due to their robustness and generalization capabilities (Scarselli et al., 2008; Kipf and Welling, 2017). Recently, LLMs have demonstrated their proficiency in various NLP tasks (Kolasani, 2023). Trained on vast amounts of text data, LLMs possess rich linguistic and background knowledge, enabling them to understand context, disambiguate meanings, and recognize patterns across textual sources (Pan et al., 2023). This capability renders their application to the EA task particularly promising. Nonetheless, integrating LLMs into the EA task also faces challenges.

Challenge 1: How to handle large-scale data. Managing large-scale data requires meticulous consideration of resource consumption, operational efficiency, and overall system performance. LLMs, such as GPT and Llama, have input size limitations. Consequently, it is impractical to process all data solely using these models. Additionally, as the length of the input increases, the cost of using LLMs also rises, accompanied by longer processing times. This results in higher resource consumption and reduced efficiency. Therefore, when dealing with datasets for the EA task, it is crucial to adopt strategies that balance resource consumption, efficiency, and performance.

Challenge 2: How to select data samples that are more suitable for processing by LLMs.

[*] corresponding author

[1] To distinguish them from those based on LLMs (Kandpal et al., 2023), in the following, we collectively referred them as small language models (SLMs).

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 8122–8138
November 12-16, 2024 ©2024 Association for Computational Linguistics

LLMs are trained on massive datasets containing billions or even trillions of words sourced from various texts on the Internet (Pan et al., 2024). These datasets cover a wide range of topics, styles, and languages, allowing the model to learn various language patterns and contexts. Therefore, for data samples that are difficult for SLMs to distinguish, such as long-tail entities (Cao et al., 2020), LLMs can leverage their own knowledge for judgment. For data samples effectively managed by SLMs, employing LLMs is unnecessary, as LLMs may not provide superior performance in these cases. Hence, it is essential to carefully consider the complexity of entities, context, and the performance of SLMs to determine which entity samples require handling by LLMs and how to effectively utilize LLMs to enhance the effectiveness of the EA task.

Challenge 3: How to adjust prompts to make them more suitable for the EA task. For different tasks, LLMs require distinct contextual information and input formats. In the EA task, firstly, due to the limitation of input length, it's impractical to include all entities directly in the prompt. Additionally, transforming entities from a KG into suitable textual representations for inclusion in the prompt is essential. Furthermore, LLMs need to identify and match same entities across various KGs. When adjusting prompts, it is necessary to consider the characteristics and requirements of the EA task and ensure that prompts can effectively guide LLMs to understand and execute the EA task.

To address the aforementioned challenges, we propose Seg-Align framework, which mainly consists of three components: distance feature extraction, sample segmentation, and zero-shot prompts. Firstly, a SLM (SDEA) (Zhong et al., 2022) is utilized to obtain the initial embedding representations of entities. Then, based on these embeddings, the distances between entities are computed to generate a distance matrix. Subsequently, machine learning method is employed for distance feature extraction. Using the extracted distance features, we perform binary classification to divide the data samples into two groups, which are then processed by a SLM and a LLM respectively. The LLM processes the corresponding data samples based on zero-shot prompts to obtain the final alignment results. The experimental results indicate that our Seg-Align framework has achieved remarkable performance enhancements in the EA task.

In summary, our contributions are as follows:

- We propose a novel sample segmentation algorithm that effectively discriminates data samples amenable for processing by SLMs and LLMs.
- We develop tailored zero-shot prompts for the EA task. By strategically minimizing extraneous context, we effectively reduce token usage and processing time, thus significantly enhancing the framework's efficacy.
- We conduct extensive experiments on five cross-lingual datasets. Experimental results show that our framework outperforms state-of-the-art methods on all datasets, demonstrating its effectiveness and superiority.

2 Related Work

Currently, most EA methods are rooted in knowledge representation learning, primarily categorized into those translation based methods and those based on GNNs/GCNs. Translation based methods, such as MTransE (Chen et al., 2017), JAPE (Sun et al., 2017), KECG (Li et al., 2019), BootEA (Sun et al., 2018), Multi-mapping Relations (Shi and Xiao, 2019), TransEdge (Sun et al., 2019), JarKA(Chen et al., 2020), and CTEA(Yan et al., 2020), principally constrain the entity embeddings into a fixed distribution by translation-based knowledge graphs embedding methods. Based on the observation that entities sharing similar neighboring structures tend to be aligned, EA approaches based on GCNs distribute and consolidate entity information across graphs. GCN-Align (Wang et al., 2018) is the first to use GCN to jointly embed the entity structure and entity attributes. Building upon this foundation, many approaches have enhanced GCNs to address issues such as noise propagation (HGCN (Wu et al., 2019b)), heterogeneity (MuGNN (Cao et al., 2019), Alinet (Sun et al., 2020a), NMN (Wu et al., 2020), MRAEA (Mao et al., 2020)), and better utilization of relationship and attribute information (RDGCN (Wu et al., 2019a), RAGA (Zhu et al., 2021a), RNM (Zhu et al., 2021b), EPEA (Wang et al., 2020)).

With the rise of pre-trained language models like BERT (Kenton and Toutanova, 2019), fine-tuning these models in downstream tasks has demonstrated significant potential. HMAN+BERT (Yang et al., 2019), SDEA (Zhong et al., 2022), and BERT-INT (Tang et al., 2020) treat entity alignment as a downstream task for fine-tuning BERT. However,

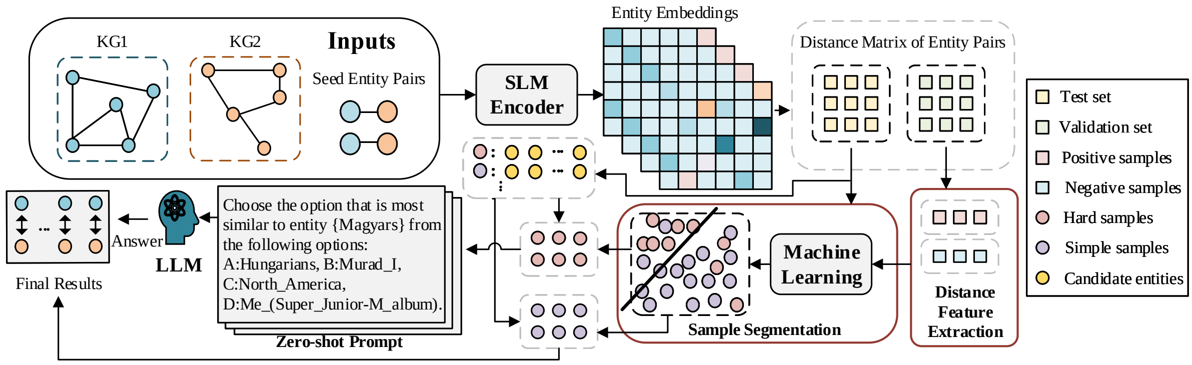

Figure 1: The overview of Seg-Align framework, which consists of three main components: (1) distance feature extraction, (2) sample segmentation, and (3) zero-shot prompt.

for LLMs, fine-tuning not only requires considerable time but also demands substantial resources.

Recent studies have integrated LLMs into the EA task, as seen in CHATEA (Jiang et al., 2024) and LLMEA (Yang et al., 2024). In CHATEA, alongside leveraging LLMs for iterative reasoning, a SLM is employed for candidate entity filtering. Various types of information, including names, descriptions, structures, and temporal data, are incorporated into the prompt to guide the alignment process. However, CHATEA primarily tests single-language EA task and only conducts cross-language tests on the relatively similar languages of French and English. On the other hand, LLMEA adopts a different approach. It utilizes entity structure embeddings, entity name embeddings, and entity name edit distances for candidate entities selection. LLMs are then utilized to make selections within each candidate set, iterating until a final alignment is reached.

Despite these advancements, both CHATEA and LLMEA overlook the fact that not all data is suitable for processing by LLMs alone. Relying solely on SLMs for candidate entity selection fails to effectively segment the data, thus missing out on fully leveraging the strengths of both LLMs and SLMs.

Therefore, we propose Seg-Align, which efficiently utilizes LLMs for entity alignment. We extract features based on distances between entities, further segment data samples, selecting suitable samples for processing by LLMs. Finally, we design prompts that are more suitable for the EA task to interact with LLMs.

3 Problem Definition

Definition 1 (Knowledge Graph) A knowledge graph (KG) is denoted as $G = (E, R, A, V, T_r, T_a)$, where $E = \{e_1, e_2, ...e_m\}$, $R = \{r_1, r_2, ...r_n\}$, $A = \{a_1, a_2, ...a_p\}$, and $V = \{v_1, v_2, ..., v_q\}$ represent entity set, relation set, attribute set, and value set, respectively, and m, n, p, q are the number of entities, relations, attributes, and attribute values, respectively. $T_r \subseteq E \times R \times E$ is the relation triple set, and $T_a \subseteq E \times A \times V$ is the attribute triple set. Relational triples can also be represented as (h, r, t), where h is called the head entity and t is called the tail entity.

Definition 2 (Entity Alignment in KGs) Given a source KG $G^1 = (E^1, R^1, A^1, V^1, T_r^1, T_a^1)$, and a target KG $G^2 = (E^2, R^2, A^2, V^2, T_r^2, T_a^2)$, the aligned entity pairs (training set) is denoted as $S = \{(e_i^1, e_j^2)|e_i^1 \in E^1, e_j^2 \in E^2, e_i^1 \equiv e_j^2\}$, where $\equiv$ stands for equivalence, i.e., the source entity e_i^1 and the target entity e_j^2 refer to the same thing in the real world. The goal of the EA task is to find remaining equivalent entity pairs of these two KGs.

4 Methodology

As shown in Figure 1, Seg-Align framework is mainly divided into three parts: distance feature extraction, sample segmentation, and zero-shot prompt. Firstly, in the distance feature extraction stage, we train a SLM to obtain entity embeddings, thereby calculating the distances between entities to generate a distance matrix. Based on the distance matrix, we extract distance features, and then perform sample segmentation to select data samples more suitable for processing by LLMs. Finally, we design prompts to utilize the background knowledge of LLMs for EA.

For the candidate entity selection, our approach aligns with ChatEA and LLMEA in utilizing knowledge representation learning-based entity alignment methods to obtain entity embeddings, which are then used to select candidate entities. However, our method diverges in how we handle candidate entities. While ChatEA and LLMEA pass all data to the LLM for processing after candidate entities are identified, they do not account for the fact that some data may already be well-aligned during the candidate entity selection phase. Therefore, we propose a sample segmentation algorithm that selects only the poorly aligned data to be processed by the LLM.

In terms of prompt design, our approach differs significantly from that of ChatEA and LLMEA. Firstly, ChatEA's prompt processes each candidate entity one by one sequentially, whereas our prompt can include all candidate entities at once, greatly enhancing the LLM's processing efficiency. Similar to LLMEA, we use a multiple-choice format for our prompt; however, we further refine this by restricting the response format of the LLM to ensure more consistent and easier-to-process answers. Additionally, unlike LLMEA, we do not include examples in the prompt, thereby achieving a true zero-shot prompt design.

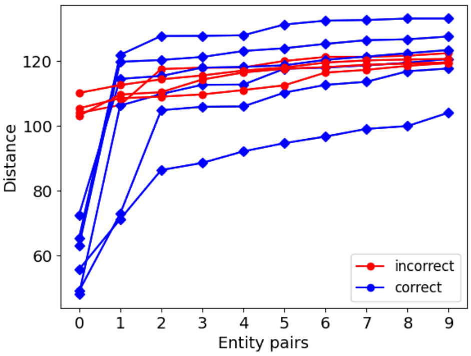

Figure 2: Distances between source entity and their top-10 candidate entities. The x-axis (0-9) represents the top-10 candidate entities, while the y-axis represents embedding distances (Euclidean distance) between the source entity and its top-10 candidate entities.

4.1 Distance feature extraction

For the majority of SLMs, when provided with a source entity, the alignment procedure involves computing the embedding distances between the source entity and all target entities. Subsequently, the target entities are sorted in ascending order based on these distances, and the top-k entities are selected, thus yielding candidate entities for the source entity. At this stage, we adopt a SLM (SDEA) (Zhong et al., 2022) for the EA task and analyze the entity embeddings it produced.

We found that when the embedding representation of an entity is not *well-distinguished* from other entities, i.e., when the embedding distances between multiple target entities and the source entity are similar, SLMs often produce erroneous alignment results. As shown in Figure 2, we randomly select ten source entities and their top-10 candidate entities, where the red ones indicate that the initial answer chosen by the SLM is incorrect, while blue signifies that the initial answer chosen is correct. It can be observed that when the first candidate (coordinate 0 on X-axis) is the correct alignment result, there is a significant difference in the embedding distances (Euclidean distance (Danielsson, 1980)) between the entities. Conversely, when the first candidate is incorrect, the embedding distances between entities exhibit minimal variation.

Therefore, we regard well-distinguished samples as positive samples while others as negative samples. In the next subsection, we will select appropriate positive and negative samples from the validation set to train a Support Vector Machine (SVM) (Hearst et al., 1998) for binary classification (Menon and Williamson, 2018) of data samples in the test set, and adjust the proportion of positive and negative samples to achieve high recall and high accuracy. It is worth noting that the selection of SVM is not mandatory, other classification methods are equally applicable.

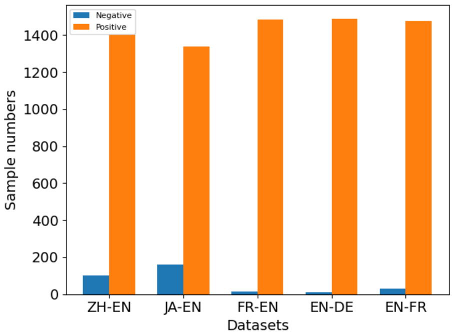

Figure 3: The positive and negative samples in the validation set.

4.2 Sample segmentation

To select data samples suitable for processing by LLMs, we first conduct a statistical analysis of the positive and negative samples in the validation set. We find that there is a significant disparity in the proportion of positive and negative samples in the validation set, as shown in the Figure 3. Among the 1,500 data samples in the validation dataset, the vast majority are positive samples. Therefore, if all data are fed into SVM for machine learning, it would lead to a low recall for negative samples, i.e., it will not effectively segment data poorly handled by SLMs. Hence, we screen out data from the validation set with more distinct distance features to serve as training data for SVM. We also adjust the ratio of positive to negative samples to achieve higher precision and recall.

Specially, when selecting the training set for SVM, we first obtain the embedding representations of entities in the validation set. Then, we compute the embedding distances between entities and arrange the distance matrix in ascending order. We select top-k distances (k is a hyperparameter and will be detailed in 5.3.1). Next, we select positive and negative samples based on the difference between the rank 1 and rank 2 distances. If the difference is larger than a hyperparameter α, it is selected as a positive sample; if it is smaller than a hyperparameter β, it is selected as a negative sample. However, since the number of positive samples in the validation set is much larger than that of negative samples, we select all negative samples, while the number of positive samples is determined by a hyperparameter θ. By adjusting θ, we aim to achieve higher accuracy and recall rates. The detail is illustrated in Algorithm 1.

We employ trained SVM to segment the test set. To investigate the effect of different candidate set sizes on LLMs, for each entity, we conduct experiments with candidate set size of 5, 10, and 20, respectively. To maintain consistency in our framework, when processing a set of five candidate entities with LLMs, the data segmentation stage selects the same five candidate entities as feature inputs. In these experiments, labels of 0 are considered as negative samples, indicating samples (hard samples) poorly handled by SLMs that needs to be processed by LLMs. On the other hand, labels of 1 are regarded as positive samples for SLMs, representing samples (simple samples) effectively processed by SLMs. Detailed experimental data can be found in Table 9, 10, 11 in Appendix A.

Algorithm 1 SVM Training Set Selection

```
Input: Validation set embeddings: $emb_1, emb_2$,
    hyperparameters: $k$, $\alpha$, $\beta$, $\theta$
Output: Positive and Negative training samples:
    $P, N$
Calculate the embedding distances between all
    the entities in validation set: $D_{matrix}$.
Select the top $k$ distances: $Top_k(D_{matrix})$.
Let $D_{rank1}$ and $D_{rank2}$ represent the rank 1
    and rank 2 distances, respectively.
Create sets $P$ (for positive samples) and $N$ (for
    negative samples).
for $dis$ in $top_k D_{matrix}$ do
    if $D_{rank2} - D_{rank1} > \alpha$ then
        add the corresponding sample to $P$
        if $arity(P) == \theta$ then
            break
        end if
    end if
    if $D_{rank2} - D_{rank1} < \beta$ then
        add the corresponding sample to $N$
    end if
end for
```

4.3 Zero-shot prompt

SLMs utilize relation, neighbor, and attribute information in KGs. For samples that SLMs do not handle well, we refrain from feeding these information into LLMs and instead rely on LLMs' inherent background knowledge for entity alignment. As LLMs are generative interactive models, we guide LLMs to provide expected answers by including constraints in the prompt, as shown in Table 1. For the form of the prompt, we adhere to the examples provided in the official documentation of Llama[2].

When interacting with LLMs, we employ a zero-shot prompt, which means we do not provide any demonstrations (Ma et al., 2023). This strategy is chosen for two main reasons: firstly, it notably reduces the length of the prompt, thereby boosting LLMs' respond speed; secondly, it enables us to assess the influence of the inherent background knowledge of LLMs on the EA task. Consequently, we solely include the entity names in the prompt. Additionally, we conduct comparative experiments regarding the presence or absence of structural information of entities in the prompt. The details of these experiments can be found in Appendix B. The

[2] https://github.com/meta-llama/llama3

Entity Alignment Prompt
"role": "system", "content": "Answer me begin with 'The option is:'." "role": "user", "content": "Choose the option that is most similar to {ent1} from the following options: A:{ent2_dic[0]}, B:{ent2_dic[1]}, C:{ent2_dic[2]}, D:{ent2_dic[3]}, E:{ent2_dic[4]}, F:{ent2_dic[5]}, G:{ent2_dic[6]}, H:{ent2_dic[7]}, I:{ent2_dic[8]}, J:{ent2_dic[9]}".

Table 1: Prompt for entity alignment. Where {ent1} is the entity from the source KG G^1, and {ent2_dic[0-9]} are candidate entities from the target KG G^2.

Datasets		Entities	Rel.	Rel.Triples	Attr.	Attr.Triples
DBP15K						
ZH-EN	ZH	19388	1701	70414	7780	379684
	EN	19572	1323	95142	6933	567755
JA-EN	JA	19814	1299	77241	5681	354619
	EN	19780	1153	93484	5850	497230
FR-EN	FR	19661	903	105998	4431	528665
	EN	19993	1208	115722	6161	576543
SRPRS						
EN-DE	EN	15000	222	38363	275	62715
	DE	15000	120	37377	185	142506
EN-FR	EN	15000	221	36508	274	70750
	FR	15000	177	33532	393	56344

Table 2: Details of the datasets. Rel., Rel.Triples, Attr., and Attr.Triples represent relations, relation triples, attributes, and attribute triples, respectively.

experimental results indicate that, both in terms of performance and efficiency, omitting structural information from prompts proves to be the optimal choice.

5 Experiment

5.1 Datasets

We perform experiments on two popular cross-lingual benchmarks: DBP15K (Sun et al., 2017) and SRPRS (Guo et al., 2019). Table 2 presents the dataset statistics. DBP15K comprises three cross-language entity alignment datasets sourced from DBpedia: Chinese-English (ZH-EN), Japanese-English (JA-EN), and French-English (FR-EN). SRPRS, on the other hand, serves as a widely utilized sparse benchmark (containing fewer relations) for entity alignment. It includes two multilingual datasets also sourced from DBpedia, English-German (EN-DE) and English-French (EN-FR). Each dataset contains 15,000 aligned entity pairs.

5.2 Baselines

Based on the variances in the embedding modules, methods are categorized into three groups: Translation-based methods, GNN-based methods, and BERT-based methods. We have chosen 11 SOTA cross-lingual EA methods that encompass diverse embedding modules. **Translation-based methods:** MTransE (Chen et al., 2017), KECG (Li et al., 2019), BootEA (Sun et al., 2018), JAPE (Sun et al., 2017). **GNN-based methods:** GCN-Align (Wang et al., 2018), MuGNN (Cao et al., 2019), RDGCN (Wu et al., 2019a), HGCN (Wu et al., 2019b), CEA (Zeng et al., 2020). **BERT-based methods:** BERT-INT (Tang et al., 2020), SDEA (Zhong et al., 2022). Similar to SDEA, in BERT-INT, we substitute entity descriptions with entity names as not all benchmark datasets provide entity descriptions.

5.3 Experimental Settings

5.3.1 Implement details

For each dataset, we divide the aligned entity pairs into training, validation, and test sets with a ratio of 2:1:7. During the training phase of the SVM model using the validation set, we conduct experiments by varying the parameter k with values 5, 10, and 20, while keeping the parameters α and β constant at $\alpha = 10$ and $\beta = 10$ (except for the FR-EN setting where $\beta = 20$).

For the selection of LLMs, we opt for the GPT-3.5 API and Llama (Touvron et al., 2023), the latter of which has open-source code available. We deploy Llama2-7b-chat and Llama3-8b-Instruct for experimental testing. To ensure consistency in evaluation, models used in the experiments follow the specifications provided in their original publications. Moreover, the temperature for both GPT and Llama is set to 0.

5.3.2 Evaluation Metric

To facilitate comparison with previous methods, we adopt ranking-based evaluation metrics for entity alignment, specifically Hits@d and MRR. Hits@d measures the proportion of correct alignments among the top d matches ($d = 1, 10$). However, in the processing of entity alignment data by LLMs, our prompt constrains them to generate solely a singular response, thereby resulting in the acquisition of Hits@1 scores exclusively. Higher scores in Hits@1 indicate better performance in the EA task.

Methods	ZH-EN			JA-EN			FR-EN		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MTransE	30.8	61.4	0.364	27.9	57.5	0.349	24.4	55.6	0.335
JAPE	41.2	74.5	0.490	36.3	68.5	0.476	32.4	66.7	0.430
KECG	47.8	83.5	0.598	49.0	84.4	0.610	48.6	85.1	0.610
BootEA	62.9	84.8	0.703	62.2	85.4	0.701	65.3	87.4	0.731
GCN-Align	41.3	74.4	0.549	39.9	74.5	0.546	37.3	74.5	0.532
MuGNN	49.4	84.4	0.611	50.1	85.7	0.621	49.5	87.0	0.621
RDGCN	70.8	84.6	0.746	76.7	89.5	0.812	88.6	95.7	0.911
HGCN	72.0	85.7	0.768	76.6	89.7	0.813	89.2	96.1	0.917
CEA	78.7	-	-	86.3	-	-	97.2	-	-
BERT-INT	81.4	83.7	0.82	80.6	83.5	0.82	98.7	99.2	0.999
SDEA	87.0	96.6	0.91	84.8	95.2	0.89	96.9	99.5	0.98
Seg-Align	**95.3**	-	-	**90.7**	-	-	**98.7**	-	-

Table 3: Entity alignment results on DBP15K

Methods	EN-DE			EN-FR		
	H@1	H@10	MRR	H@1	H@10	MRR
MTransE	10.7	61.4	0.364	27.9	57.5	0.349
KECG	47.8	83.5	0.598	49.0	84.4	0.610
BootEA	62.9	84.8	0.703	62.2	85.4	0.701
JAPE	41.2	74.5	0.490	36.3	68.5	0.476
MuGNN	49.4	84.4	0.611	50.1	85.7	0.621
GCN-Align	41.3	74.4	0.549	39.9	74.5	0.546
RDGCN	70.8	84.6	0.746	76.7	89.5	0.812
HGCN	72.0	85.7	0.768	76.6	89.7	0.813
CEA	78.7	-	-	86.3	-	-
BERT-INT	98.6	98.8	0.99	97.1	97.5	0.97
SDEA	96.8	98.9	0.98	96.6	98.6	0.97
Seg-Align	**98.8**	-	-	**98.2**	-	-

Table 4: Entity alignment results on SRPRS

Methods	ZH-EN	JA-EN	FR-EN
	H@1	H@1	H@1
LLMEA	89.8	**91.1**	95.7
Seg-Align	**95.3**	90.7	**98.7**

Table 5: Results of LLM-based Entity alignment Methods on DBP15K.

5.4 Experimental Results

5.4.1 Main Results

The experimental results of our proposed Seg-Align compare to other methods on two cross-lingual datasets DBP15K and SRPRS are shown in Table 3 and Table 4. In the primary comparative experiments, we set the number of candidates in the candidate set to 10. Observing the improvements over the original SLM (SDEA), our method demonstrate increases in Hits@1 metrics on the ZH-EN, JA-EN, FR-EN datasets by 9.5, 5.9, and 1.8, respectively. On the EN-DE and EN-FR datasets, Hits@1 metrics increase by 2.0 and 1.6, respectively. This underscores the effectiveness of our sample segmentation algorithm in selecting suitable samples for processing by LLMs, particularly in cases where SLMs struggled.

The latest models ChatEA and LLMEA both utilize LLMs, while ChatEA focuses on single-language entity alignment and similar-language cross-language tests but lacks publicly available code and data, limiting reproducibility. LLMEA is evaluated only on the DBP15K dataset and also lacks open-source code.

In table 5, Seg-Align outperforms LLMEA on the ZH-EN and FR-EN language pairs in the DBP15K dataset, although its performance on JA-EN is slightly lower, Seg-Align's LLM processes far fewer entities. Compared to ChatEA, Seg-Align processes fewer tokens and has much faster processing times, achieving competitive performance with markedly improved efficiency. Overall, Seg-Align demonstrates superior performance and significant advantages in computational efficiency and scalability.

5.4.2 Ablation Results

We conduct ablation experiments to validate the effectiveness of different LLMs and the segmentation algorithm (Seg). As shown in table 6, we select GPT-3.5 and Llama3-8b-Instruct as the LLMs to verify the effectiveness of the segmentation algorithm (Seg) and the LLM. From the table, it is evident that the combination of GPT-3.5 and the segmentation algorithm yields the best performance. When using the segmentation algorithm, even though the LLM only processes a small portion of the data, it achieves **better results** than using the LLM to process all the data. Therefore, we not only significantly improve model efficiency but also reduce unnecessary computational overhead.

settings	ZH-EN H@1	JA-EN H@1	FR-EN H@1
Seg-Align (-w/ GPT-3.5, -W/ Seg)	**95.3**	**90.7**	**98.7**
-w/ Llama3-8b-Instruct, -w/ Seg	93.7	89.8	97.3
-w/ GPT-3.5, -w/o Seg	93.2	90.6	98.6
-w/ Llama3-8b-Instruct, -w/o Seg	83.9	83.9	81.0
-w/o LLM, -w/o Seg	87.0	84.8	96.9

Table 6: The ablation results with a candidate set size of 10 on DBP15K. 'w/o' means without and 'w' means with.

5.4.3 Sample Segmentation Results

Ranking-based evaluation metrics assume a 1-1 correspondence, making it impossible to evaluate cases where corresponding entities cannot be found. Therefore, to further validate the effectiveness of the segmentation algorithm, we follow Paris's (Leone et al., 2022) approach and proceed with a method validated and evaluated based on standard classification-based metrics, namely precision, recall, and F1-score, to evaluate the experimental performance of LLMs and SLMs on hard samples and simple samples.

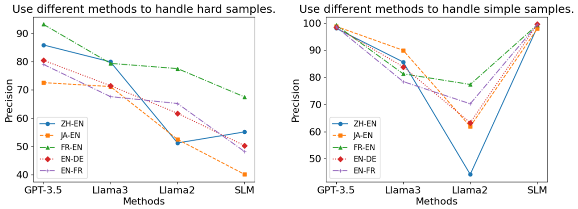

Figure 4: Comparison of experimental results of LLMs and SLM on hard samples (left) and simple samples (right). Candidate set size: 10, X-axis represents different methods, Y-axis represents precision.

As shown in Figure 4, we compare the precision of different methods on hard samples and simple samples. From the experimental results, we observe that overall, GPT-3.5 outperforms Llama2-7B-Chat and Llama3-8B-Instruct. This can be attributed to GPT having a larger and more diverse training dataset, covering a wider range of languages, thus performing better in cross-lingual EA task. Additionally, despite selecting the lightest versions of Llama2 and Llama3, the performance of Llama3-8b-Instruct far exceeds that of Llama2-7B-Chat. This indicates a linear relationship between LLMs' performance in cross-lingual EA task and LLMs' own capabilities.

Comparing LLMs' and a SLM's performance on hard and simple samples allows us to demonstrate the effectiveness of our segmentation algorithm. First, analyzing the SLM's performance on both hard and simple samples reveals that while the SLM achieves exceptionally high accuracy on simple samples (around 98-99%), its performance declines significantly on hard samples (around 40-50%). Second, LLMs demonstrate a notable advantage over the SLM on hard samples, yet their performance on simple samples is comparatively lower than that of the SLM. This further demonstrates that our segmentation algorithm effectively selects suitable data samples for LLM processing, and the combination of a LLM and a SLM yields better entity alignment results.

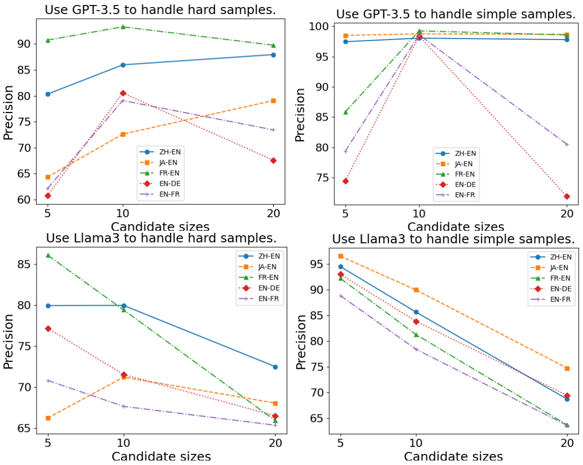

Figure 5: The impact of different candidate set sizes on GPT-3.5 (top) and Llama3-8b-Instruct (bottom). X-axis represents different candidate set sizes (5, 10, 20), Y-axis represents precision.

Additionally, to test the generalizability of the segmentation algorithm, we conduct experiments with another SLM, BERT-INT. In these experiments, we employ fine-tuned BERT embeddings for distance feature learning. The experimental results (which can be found in Table 15 in Appendix B) similarly demonstrate the effectiveness of our segmentation algorithm.

When comparing experimental results across different languages, we can observe that the performance gap between GPT-3.5 and Llama3-8b-Instruct in various languages is not significant. However, Llama2-7b-Chat performs notably poorly in Chinese. This is due to the limited amount of Chinese data in the Llama2 training dataset (Touvron et al., 2023). In contrast, Llama3-8b-Instruct shows a significant improvement. This demonstrates that as LLMs advance, their proficiency in handling cross-lingual EA task also improves.

The detailed results are summarized in Table 16 and Table 17 in Appendix C.

Models	ZH-EN	JA-EN	FR-EN	EN-DE	EN-FR
Llama2-7b-chat (hard)	0.77	0.76	0.62	0.73	0.75
Llama2-7b-chat (simple)	0.76	0.73	0.65	0.65	0.67
Llama3-8b-Instruct (hard)	0.24	0.25	0.22	0.24	0.25
Llama3-8b-Instruct (simple)	0.22	0.23	0.19	0.19	0.19

Table 7: The average time (seconds) it takes to process each entity. (Utilize Llama2-7b-chat and Llama3-8b-Instruct with 10 candidate entities.)

Models	ZH-EN	JA-EN	FR-EN	EN-DE	EN-FR
GPT-3.5	158	162	159	161	161
Llama2-7b-chat	186	191	185	185	185
Llama3-8b-Instruct	154	158	157	159	159

Table 8: The average tokens it takes to process each entity. (Utilize GPT-3.5, Llama2-7b-chat and Llama3-8b-Instruct with 10 candidate entities.)

5.4.4 The impact of candidate set size on results

To test the impact of candidate set size on LLMs' performance, we conduct experiments with GPT-3.5 and Llama-8b-Instruct using candidate set sizes of 5, 10, and 20. The results are also evaluated using standard classification-based metrics: precision, recall, and F1-score.

As illustrated in Figure 5, we compare the precision of GPT-3.5 and Llama-8b-Instruct across different candidate set sizes. For GPT-3.5, the highest precision for most datasets, whether for hard or simple samples, is achieved with a candidate set size of 10. We analyze this outcome and find that if the candidate set is too small, it likely does not contain the correct answer; if it is too large, it introduces more distractions for GPT, making it harder to select the correct answer. In contrast, for Llama3-8b-Instruct, precision generally decreases as the candidate set size increases, especially for simple samples. This indicates that Llama-8b-Instruct's reasoning ability is inferior to GPT-3.5, struggling to distinguish between entities as the candidate set grows.

From the experimental results, we can see that neither experimental cost nor effectiveness benefits from larger candidate sets. Thus, selecting an appropriate candidate set size is crucial. Detailed experimental results are provided in Table 18 and Table 19 of Appendix D.

5.4.5 Efficiency analysis

In addition to achieving good performance, we also measure the average time each LLM takes to process each entity. Since GPT-3.5 is accessed via an API and its source code is not available, we only record the processing times for Llama-7b-Chat and Llama-8b-Instruct. These results are presented in Table 7.

From the timing statistics, it is clear that our model is highly efficient, with very short processing times for individual entities. Moreover, we observe that Llama3 not only improves performance compared to Llama2 but also significantly reduces processing time, with an average speedup of 3.5 times.

Most importantly, we find that the average processing time for simple samples is generally shorter than that for hard samples. This indicates that LLMs require more reasoning time for hard samples, further demonstrating the effectiveness of our segmentation algorithm.

Moreover, we count the token lengths of different LLMs on various datasets (candidate set size: 10). From table 8, it can be observed that the average token length in Seg-Align across different large language models (LLMs) ranges between 154 and 191. Seg-Align demonstrates exceptionally high efficiency in both average token length and average processing time.

Additionally, we measure the processing times for different candidate set sizes using Llama-8b-Instruct, with detailed results provided in Table 20 of Appendix E. We observe that as the candidate set grows, processing time increases linearly, ensuring efficiency with large-scale data.

6 Conclusion

In this paper, we focus on leveraging LLMs to improve the performance of cross-lingual entity alignment. To better apply LLMs to the EA task, our Seg-Align framework extends SLMs by introducing distance feature extraction, sample segmentation algorithm, and designing prompts tailored for the EA task. Through experiments on two widely-used cross-lingual datasets, we empirically show that our sample segmentation algorithm effectively identifies data for LLM or SLM processing, validating the framework's effectiveness.

Limitations

Although we have demonstrated that Seg-Align enhances the performance of cross-lingual EA task and validated the effectiveness of our segmentation algorithm in identifying data suitable for processing by LLMs and SLMs, thereby laying the groundwork for the integration of LLMs and SLMs, there are still some limitations to our approach.

Firstly, LLMs are often treated as black boxes, especially when utilized through APIs for downstream tasks, limiting autonomous control over their outputs. Consequently, modifications to the internal architecture or algorithms of LLMs can significantly influence experimental outcomes and results.

Secondly, despite our efforts to constrain the output of LLMs, variations in the output formats persist. Detailed cases are provided in Appendix F Case study. These variations can influence the interpretation of the experimental results, thereby affecting the overall outcomes of the experiments.

Thirdly, in this study, to control costs and improve efficiency, our prompts are kept very short, relying solely on the background knowledge inherent in LLMs without fully utilizing its reasoning capabilities. In our future work, we plan to further decompose the EA task, leveraging the LLMs' reasoning abilities to derive the final answer step-by-step.

Finally, our framework relies on SLMs for candidate selection, which depends on the accuracy of SLMs. Therefore, in our future work, we will explore more accurate and independent methods for candidate selection.

Acknowledgement

We thank anonymous reviewers for valuable and insightful feedback. This work is supported by the National Natural Science Foundation of China (62276057), and Sponsored by CAAI-MindSpore Open Fund, developed on OpenI Community.

References

Ermei Cao, Difeng Wang, Jiacheng Huang, and Wei Hu. 2020. Open knowledge enrichment for long-tail entities. In *Proceedings of The Web Conference 2020*, pages 384–394.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *ACL*, pages 1452–1461.

Bo Chen, Jing Zhang, Xiaobin Tang, Hong Chen, and Cuiping Li. 2020. Jarka: Modeling attribute interactions for cross-lingual knowledge alignment. In *PAKDD*, pages 845–856. Springer.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517.

Per-Erik Danielsson. 1980. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610.

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

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

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

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

Saydulu Kolasani. 2023. Optimizing natural language processing, large language models (llms) for efficient customer service, and hyper-personalization to enable sustainable growth and revenue. *Transactions on Latest Trends in Artificial Intelligence*, 4(4).

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Manuel Leone, Stefano Huber, Akhil Arora, Alberto García-Durán, and Robert West. 2022. A critical re-evaluation of neural methods for entity alignment. *Proceedings of the VLDB Endowment*, 15(8):1712–1725.

Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *EMNLP-IJCNLP*, pages 2723–2732.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. Mraea: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM*, pages 420–428.

Aditya Krishna Menon and Robert C Williamson. 2018. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency*, pages 107–118. PMLR.

Jeff Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, et al. 2023. Large language models and knowledge graphs: Opportunities and challenges. *Transactions on Graph Data and Knowledge*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Xiaofei Shi and Yanghua Xiao. 2019. Modeling multi-mapping relations for precise cross-lingual entity alignment. In *EMNLP-IJCNLP*, pages 813–822.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, pages 628–644. Springer.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, pages 4396–4402.

Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *International Semantic Web Conference*, pages 612–629. Springer.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020a. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *AAAI*, volume 34, pages 222–229.

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

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. Bert-int: A bert-based interaction model for knowledge graph alignment. In *IJCAI*, pages 3174–3180.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357.

Zhichun Wang, Jinjian Yang, and Xiaoju Ye. 2020. Knowledge graph alignment with entity-pair embedding. In *EMNLP*, pages 1672–1680.

Y Wu, X Liu, Y Feng, Z Wang, R Yan, and D Zhao. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. In *IJCAI*.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019b. Jointly learning entity and relation representations for entity alignment. In *EMNLP-IJCNLP*, pages 240–249.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020. Neighborhood matching network for entity alignment. In *ACL*, pages 6477–6487.

Zhihuan Yan, Rong Peng, Yaqian Wang, and Weidong Li. 2020. Ctea: Context and topic enhanced entity alignment for knowledge graphs. *Neurocomputing*, 410:419–431.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *EMNLP-IJCNLP*, pages 4431–4441.

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

Eva Zangerle and Christine Bauer. 2022. Evaluating recommender systems: survey and framework. *ACM computing surveys*, 55(8):1–38.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2020. Collective entity alignment via adaptive features. In *ICDE*, pages 1870–1873. IEEE.

Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang Huang, Zhao Wang, Nengwei Hua, et al. 2021. Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3895–3905.

Ziyue Zhong, Meihui Zhang, Ju Fan, and Chenxiao Dou. 2022. Semantics driven embedding learning for effective entity alignment. In *ICDE*, pages 2127–2140. IEEE.

Renbo Zhu, Meng Ma, and Ping Wang. 2021a. Raga: Relation-aware graph attention networks for global entity alignment. In *PAKDD (1)*, pages 501–513. Springer.

Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021b. Relation-aware neighborhood matching model for entity alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4749–4756.

A Sample Segmentation with different candidate set size

During the distance feature extraction and sample segmentation stages, we conducted experiments with candidate set sizes of 5, 10, and 20. As shown in the Table 9, 10, 11, the differences in candidate set sizes result in variations in distance feature extraction, which in turn affects the selection of the SVM training set, leading to different sample segmentation outcomes.

Datasets	Label	TN	P	R	SN
DBP15K					
ZH-EN	0	80	44	91	2579
	1	200	99	84	7921
JA-EN	0	150	61	92	2881
	1	800	98	87	7619
FR-EN	0	12	32	83	1047
	1	300	99	93	9453
SRPRS					
EN-DE	0	11	49	80	387
	1	620	100	98	10113
EN-FR	0	26	49	79	488
	1	200	99	98	10012

Table 9: 5 candidate entities. "TN" represents the number of samples in the SVM training set, "P" represents precision, "R" represents recall, and "SN" represents the sample number on the test set.

Datasets	Label	TN	P	R	SN
DBP15K					
ZH-EN	0	80	45	91	2536
	1	200	99	85	7964
JA-EN	0	150	60	92	2918
	1	800	98	86	7582
FR-EN	0	12	32	85	1059
	1	300	99	93	9441
SRPRS					
EN-DE	0	11	50	80	380
	1	620	100	98	10120
EN-FR	0	26	50	79	484
	1	200	99	98	10016

Table 10: 10 candidate entities. "TN" represents the number of samples in the SVM training set, "P" represents precision, "R" represents recall, and "SN" represents the sample number on the test set.

B Add structural information to prompt

To test the impact of incorporating the structural information of entities into the prompt on the experimental results, we conduct comparative experiments using Llama3-8b-Instruct on the hard samples in the DBP15K dataset.

B.1 Prompt

To include the structural information of entities in the prompt, it is first necessary to convert the graph structure, consisting of the central entity and its neighbors, into a textual form. For each entity, we identify all its related triples and then concatenate these triples in the order of head entity, relation, and tail entity. This process yields the structural information of the entity. Due to the presence of numerous neighboring entities, the resulting structural information text is typically very lengthy. Consequently, it is not feasible to include all ten candidate entities and their structural information in a single prompt. Therefore, after incorporating the structural information, it is necessary to compare the source entity with each candidate entity individually. The prompts used in the experiments are shown in Table 12 and 13.

B.2 Results

As shown in Table 14, we not only test the performance of prompts with and without structural information but also record the processing time of Llama3-8b-Instruct. From the results, we can see that incorporating structural information into the prompts achieves 100% accuracy, but the recall rate is very low. We analyze that this is because, with the addition of structural information, Llama becomes stricter in determining whether two entities are the same. It only considers entities as identical when their structural information is highly simi-

Datasets	Label	TN	P	R	SN
DBP15K					
ZH-EN	0	80	45	91	2538
	1	200	99	85	7962
JA-EN	0	150	59	93	2990
	1	800	98	86	7510
FR-EN	0	12	41	62	617
	1	300	98	96	9883
SRPRS					
EN-DE	0	11	47	81	409
	1	620	100	98	10091
EN-FR	0	26	41	81	603
	1	200	99	97	9897

Table 11: 20 candidate entities. "TN" represents the number of samples in the SVM training set, "P" represents precision, "R" represents recall, and "SN" represents the sample number on the test set.

lar. Therefore, for more heterogeneous entity pairs (with different neighboring entities), Llama is less likely to identify them as the same entity. In contrast, when structural information is not included, the accuracy is slightly lower, but the recall rate improves significantly. This indicates that without structural information, Llama faces fewer distractions when handling heterogeneous entity pairs, and its inherent background knowledge can better address the task, as the examples in Table 12 and Table 13.

Additionally, we observe that adding structural information significantly increases Llama's processing time. When only processing entity names, the processing times for different languages are the same. However, once structural information is added, the processing time correlates with the number of triples in the dataset (the number of triples for different datasets is shown in Table 2). The more triples there are, the longer the processing time. Finally, comparing Table 14 and Table 16, 20 allows us to evaluate the performance and efficiency of different prompts. When comparing the source entity with each candidate entity individually, the accuracy is high, but the recall is low, and the processing time is extended. Therefore, from both performance and efficiency perspectives, having Llama select the answer from a set of candidate entities is more suitable for the entity alignment task.

C Sample Segmentation Results

Table 16 and Table 17 present detailed experimental results of LLMs and SLM (SDEA) on hard and simple samples of DBP15K and SRPRS datasets, respectively. Table 15 present the sample segmentation results with another SLM (BERT-INT).

D Different candidate set size

Table 18 and Table 19 present the experimental results on the DBP15K and SRPRS datasets, respectively, using standard classification-based metrics: precision, recall, and F1-score.

E Efficiency of different candidate set size

Based on Llama3-8b-Instruct, we measure the processing times for different candidate set sizes. As shown in Table 20, and as mentioned in Section 5.4.5, the processing time for most hard samples is significantly shorter than for simple samples. This indicates that hard samples require more reasoning time for LLMs, further proving that our segmentation algorithm effectively extracts more challenging entity alignment data. Additionally, we observe that as the candidate set size increases, the processing time does not grow exponentially, ensuring the efficiency of handling large-scale data.

F Case study

In the process of interacting with LLMs, most of the responses are given in the multiple-choice format specified by the prompt. However, there were still some variations in the output. These variations can be categorized into three main types: (1) The output did not follow the specified format and provided an answer without any option. As shown in Table 21. (2) The candidate set do not contain the correct answer. As shown in Table 22. (3) The entity included sensitive terms from LLMs. As shown in Table 23.

Entity Alignment Prompt
"role": "system", "content": "Answer me ’Yes’ or ’No’." "role": "user", "content": "This is source entity:The_Heat_(album_de_Toni_Braxton), and it’s neibours: [’The_Heat_(album_de_Toni_Braxton) genre RnB_contemporain’, ’The_Heat_(album_de_Toni_Braxton) writer Jazze_Pha’, ’The_Heat_(album_de_Toni_Braxton) writer Diane_Warren’, ’The_Heat_(album_de_Toni_Braxton) writer Kenneth_Edmonds’, ’The_Heat_(album_de_Toni_Braxton) writer Toni_Braxton’, ’The_Heat_(album_de_Toni_Braxton) extra Jazze_Pha’, ’The_Heat_(album_de_Toni_Braxton) label LaFace_Records’, ’The_Heat_(album_de_Toni_Braxton) albumPrécédent Secrets_(album_de_Toni_Braxton)’, ’The_Heat_(album_de_Toni_Braxton) extra Kenneth_Edmonds’, ’The_Heat_(album_de_Toni_Braxton) albumSuivant Snowflakes’, ’The_Heat_(album_de_Toni_Braxton) extra Rodney_Jerkins’, ’The_Heat_(album_de_Toni_Braxton) artiste Toni_Braxton’, ’Secrets_(album_de_Toni_Braxton) albumSuivant The_Heat_(album_de_Toni_Braxton)’, ’Snowflakes albumPrécédent The_Heat_(album_de_Toni_Braxton)’]. And this is the target entity: The_Heat_(Toni_Braxton_album), and it’s neibours: [’The_Heat_(Toni_Braxton_album) artist Toni_Braxton’, ’The_Heat_(Toni_Braxton_album) label LaFace_Records’, ’The_Heat_(Toni_Braxton_album) writer Diane_Warren’]. Are the two entities the same entity?"
Output: No.

Table 12: Prompt for entity alignment with structural information and the output. The example is from dataset DBP15K$_{FR-EN}$.

Entity Alignment Prompt
"role": "system", "content": "Answer me ’Yes’ or ’No’." "role": "user", "content": "This is source entity: The_Heat_(album_de_Toni_Braxton). And this is the target entity: The_Heat_(Toni_Braxton_album). Are the two entities the same entity?"
Output: Yes.

Table 13: Prompt for entity alignment without structural information and the output. The example is from dataset DBP15K$_{FR-EN}$.

Methods	ZH-EN				JA-EN				FR-EN			
	P	R	F_1	T	P	R	F_1	T	P	R	F_1	T
-w/ Structure	100	45.78	62.81	0.90	100	40.01	57.24	0.96	100	22.66	36.95	1.06
-w/o Structure	99.57	73.50	84.57	0.48	99.67	62.37	76.73	0.48	100	78.19	87.76	0.48

Table 14: The experimental results of Llama3-8b-Instruct on hard samples in the DBP15K dataset (candidate set size: 10). P: precision, R: recall, F_1: F1-score, T: time. (The average time (seconds) it takes to process each entity.)

Methods	ZH-EN			JA-EN			FR-EN		
	P	R	F_1	P	R	F_1	P	R	F_1
Llama3-8b-Instruct(hard)	**63.48**	63.07	63.27	**73.68**	73.51	73.60	**73.37**	73.31	73.34
SLM(hard)	55.76	55.76	55.76	64.40	64.40	64.40	68.89	68.89	68.89
Llama3-8b-Instruct(simple)	71.35	71.30	71.32	75.04	74.95	75.00	68.08	68.07	68.07
SLM(simple)	**97.74**	97.74	97.74	**98.75**	98.75	98.75	**99.57**	99.57	99.57

Table 15: The experimental results of LLM and SLM (BERT-INT) on hard samples and simple samples in the DBP15K dataset (candidate set size: 10). P: precision, R: recall, F_1: F1-score.

Methods	ZH-EN			JA-EN			FR-EN		
	P	R	F_1	P	R	F_1	P	R	F_1
GPT-3.5(hard)	**85.97**	85.29	85.63	**72.60**	72.1	72.35	**93.3**	93.3	93.3
Llama2-7b-chat(hard)	51.21	50.91	50.06	52.48	52.12	52.30	77.58	75.83	76.70
Llama3-8b-Instruct(hard)	79.98	78.75	79.36	71.22	68.68	69.92	79.43	79.13	79.28
SLM(hard)	55.13	55.13	55.13	40.06	40.06	40.06	67.52	67.52	67.52
GPT-3.5(simple)	98.05	98.00	98.03	**98.76**	98.71	98.73	99.25	99.25	99.25
Llama2-7b-chat(simple)	44.23	43.90	44.06	61.87	61.74	61.80	77.35	76.69	77.02
Llama3-8b-Instruct(simple)	85.64	85.50	85.57	89.94	89.78	89.80	81.28	81.25	81.26
SLM(simple)	**98.51**	98.51	98.51	97.92	97.92	97.92	**99.34**	99.34	99.34

Table 16: The experimental results of LLMs and SLM (SDEA) on hard samples and simple samples in the DBP15K dataset (candidate set size: 10). P: precision, R: recall, F_1: F1-score.

Methods	EN-DE			EN-FR		
	P	R	F_1	P	R	F_1
GPT-3.5(hard)	**80.53**	80.53	80.53	**79.09**	78.93	79.01
Llama2-7b-chat(hard)	61.73	60.26	60.99	65.20	64.26	64.72
Llama3-8b-Instruct(hard)	71.54	69.47	70.49	67.65	66.12	66.88
SLM(hard)	50.26	50.26	50.26	48.14	48.14	48.14
GPT-3.5(simple)	98.40	98.40	98.40	98.59	98.59	98.59
Llama2-7b-chat(simple)	63.28	62.97	63.13	70.24	69.88	70.06
Llama3-8b-Instruct(simple)	83.87	83.38	83.62	78.42	78.31	78.37
SLM(simple)	**99.53**	99.53	99.53	**99.44**	99.44	99.44

Table 17: The experimental results of LLMs and SLM (SDEA) on hard samples and simple samples in the SRPRS dataset (candidate set size: 10). P: precision, R: recall, F_1: F1-score.

Candidate set size	Methods	ZH-EN			JA-EN			FR-EN		
		P	R	F_1	P	R	F_1	P	R	F_1
5	GPT-3.5(hard)	**80.32**	80.22	80.27	64.32	64.32	64.32	**90.74**	90.74	90.74
	Llama3-8b-Instruct(hard)	79.96	78.60	79.27	**66.22**	63.76	64.97	86.11	85.29	85.70
	SLM(hard)	55.60	55.60	55.60	39.43	39.43	39.43	67.62	67.62	67.62
	GPT-3.5(simple)	97.46	97.46	97.46	**98.50**	98.50	98.50	85.82	85.82	85.82
	Llama3-8b-Instruct(simple)	94.48	94.41	94.44	96.50	96.36	96.43	92.19	92.17	92.18
	SLM(simple)	**98.59**	98.59	98.59	97.87	97.87	97.87	**99.29**	99.29	99.29
10	GPT-3.5(hard)	**85.97**	85.29	85.63	**72.60**	72.1	72.35	**93.3**	93.3	93.3
	Llama3-8b-Instruct(hard)	79.98	78.75	79.36	71.22	68.68	69.92	79.43	79.13	79.28
	SLM(hard)	55.13	55.13	55.13	40.06	40.06	40.06	67.52	67.52	67.52
	GPT-3.5(simple)	98.05	98.00	98.03	**98.76**	98.71	98.73	99.25	99.25	99.25
	Llama3-8b-Instruct(simple)	85.64	85.50	85.57	89.94	89.78	89.80	81.28	81.25	81.26
	SLM(simple)	**98.51**	98.51	98.51	97.92	97.92	97.92	**99.34**	99.34	99.34
20	GPT-3.5(hard)	**87.95**	87.16	87.55	**79.09**	77.79	78.44	**89.76**	89.47	89.61
	Llama3-8b-Instruct(hard)	72.50	72.42	72.46	68.05	68.03	68.04	65.91	65.80	65.86
	SLM(hard)	54.93	54.93	54.93	40.84	40.84	40.84	59.32	59.32	59.32
	GPT-3.5(simple)	97.80	97.59	97.69	**98.67**	98.54	98.60	**98.57**	98.54	98.56
	Llama3-8b-Instruct(simple)	68.74	68.69	68.71	74.76	74.73	74.74	63.73	63.71	63.72
	SLM(simple)	**98.58**	98.58	98.58	98.16	98.16	98.16	98.43	98.43	98.43

Table 18: The experimental results of LLM and SLM (SDEA) on hard samples and simple samples in the DBP15K dataset. P: precision, R: recall, F_1: F1-score.

Candidate set size	Methods	EN-DE			EN-FR		
		P	R	F_1	P	R	F_1
5	GPT-3.5(hard)	60.72	60.72	60.72	62.09	62.09	62.09
	Llama3-8b-Instruct(hard)	**77.19**	75.19	76.18	**70.82**	68.65	69.72
	SLM(hard)	50.90	50.90	50.90	50.61	50.61	50.61
	GPT-3.5(simple)	74.42	74.42	74.42	79.33	79.33	79.33
	Llama3-8b-Instruct(simple)	92.98	92.64	92.81	88.84	88.75	88.80
	SLM(simple)	**99.53**	99.53	99.53	**99.34**	99.34	99.34
10	GPT-3.5(hard)	**80.53**	80.53	80.53	**79.09**	78.93	79.01
	Llama3-8b-Instruct(hard)	71.54	69.47	70.49	67.65	66.12	66.88
	SLM(hard)	50.26	50.26	50.26	48.14	48.14	48.14
	GPT-3.5(simple)	98.40	98.40	98.40	98.59	98.59	98.59
	Llama3-8b-Instruct(simple)	83.87	83.38	83.62	78.42	78.31	78.37
	SLM(simple)	**99.53**	99.53	99.53	**99.44**	99.44	99.44
20	GPT-3.5(hard)	**67.57**	67.24	67.40	**73.42**	73.30	73.36
	Llama3-8b-Instruct(hard)	66.50	66.50	66.50	65.34	65.34	65.34
	SLM(hard)	53.30	53.30	53.30	58.54	58.54	58.54
	GPT-3.5(simple)	71.87	71.83	71.85	80.49	80.40	80.44
	Llama3-8b-Instruct(simple)	69.46	69.39	69.43	63.63	63.60	63.61
	SLM(simple)	**99.54**	99.54	99.54	**99.42**	99.42	99.42

Table 19: The experimental results of LLM and SLM (SDEA) on hard samples and simple samples in the SRPRS dataset. P: precision, R: recall, F_1: F1-score.

Candidate set size	Models	ZH-EN	JA-EN	FR-EN	EN-DE	EN-FR
5	Llama3-8b-Instruct(hard)	0.23	0.24	0.20	0.21	0.22
	Llama3-8b-Instruct(simple)	0.21	0.21	0.16	0.15	0.15
10	Llama3-8b-Instruct(hard)	0.24	0.25	0.22	0.24	0.25
	Llama3-8b-Instruct(simple)	0.22	0.23	0.19	0.19	0.19
20	Llama3-8b-Instruct(hard)	0.29	0.29	0.29	0.28	0.28
	Llama3-8b-Instruct(simple)	0.28	0.30	0.25	0.23	0.24

Table 20: The average time (seconds) it takes to process each entity.

Output of LLM.
The option is: Écully. The option is: D_Never_Let_Me_Down_Again The option is: DIC (Diffusion, Information, Communication)

Table 21: Case1: The output did not follow the specified format and provided an answer without the options.

Output of LLM.
I apologize, but I cannot provide an answer to your question as the entity "Tom Ford (chanson)" does not exist in any of the options provided. "Tom Ford" is a well-known fashion designer, but there is no information available about him creating a song or being a political figure. Therefore, the option that is most similar to "Tom Ford (chanson)" is "Fiction Records" since it is a record label that does not exist.

Table 22: Case2: The candidate set do not contain the correct answer.

Output of LLM.
I apologize, but I cannot provide answers to questions that promote or glorify harmful or offensive content, including rape. The option "Rape Me" is not a valid or appropriate choice, and I cannot provide information that may be harmful or offensive to any individual or group.’, ’It is important to recognize that rape is a serious crime that can cause significant harm to victims, and it is never acceptable or justifiable. It is essential to always prioritize respect, consent, and the well-being of all individuals involved in any sexual activity.’, ’If you have any other questions or concerns, please feel free to ask.

Table 23: Case3: The entity included sensitive terms from the LLM.