Match, Compare, or Select? An Investigation of Large Language Models for Entity Matching

Anonymous ACL submission

Abstract

Entity matching (EM) is a critical step in entity resolution. Recently, entity matching based on large language models (LLMs) has shown great promise. However, current LLM-based entity matching approaches typically follow a binary matching paradigm that ignores the global consistency between record relationships. In this paper, we investigate various methodologies for LLM-based entity matching that incorporate record interactions from different perspectives. Specifically, we comprehensively compare three representative strategies: matching, comparing, and selecting, and analyze their respective advantages and challenges in diverse scenarios. Based on our findings, we further design a compound entity matching framework (ComEM) that leverages the composition of multiple strategies and LLMs. ComEM benefits from the advantages of different sides and achieves improvements in both effectiveness and efficiency. Experimental results verify that ComEM not only achieves significant performance gains on various datasets, but also reduces the cost of LLM-based entity matching for practical applications.

1 Introduction

Entity resolution (ER), also known as record linkage (Fellegi and Sunter, 1969) or deduplication (Elmagarmid et al., 2007), aims to identify and merge records that refer to the same real-world entity. Entity matching (EM) is a critical step in entity resolution that uses complex techniques to identify matching records from candidate pairs filtered by the blocking step (Papadakis et al., 2021). The recent emergence of large language models (LLMs) has introduced a new zero- or few-shot paradigm to EM, showing great promise (Narayan et al., 2022; Peeters and Bizer, 2023b,a; Fan et al., 2023; Li et al., 2024). LLM-based entity matching methods can achieve similar or even better performance than deep learning methods trained on large amounts of data, and are less susceptible to the unseen entity problem (Wang et al., 2022; Peeters et al., 2024).

Do these two records refer to the same real-world entity?
(1) Title: *Cruzer Force USB Flash Drive 32GB Type-A 2.0 Chrome*
(2) Title: *Sandisk USB Flash Drive 32GB Cruzer Glide 2.0/3.0*
LLM Response **No** (a) Matching

Which of these two records is more consistent with the given record:
Title: *Cruzer Force USB Flash Drive 32GB Type-A 2.0 Chrome*
(A) Title: *Sandisk USB Flash Drive 32GB Cruzer Glide 2.0/3.0*
(B) Title: *Pendrive Sandisk Cruzer Force - SDCZ71-032G-B35*
LLM Response **Record B** (b) Comparing

Select a record from the following list that refers to the same real-world entity as the given record:
Title: *Cruzer Force USB Flash Drive 32GB Type-A 2.0 Chrome*
(1) Title: *Sandisk USB Flash Drive 32GB Cruzer Glide 2.0/3.0*
(2) Title: *Pendrive Sandisk Cruzer Force - SDCZ71-032G-B35*
(3) Title: *Sandisk Extreme Pro 3.1 Solid State Flash Drive 128GB*
(4) Title: *Kingston DataTraveler G4 32 GB USB-stick*
...
LLM Response **Record 2** (c) Selecting

Figure 1: Three strategies for LLM-based entity matching. We omit other attributes of records for simplicity.

However, current LLM-based entity matching methods identify matches by classifying each pair of records independently. This *matching* strategy ignores the global consistency between record relationships and thus leads to suboptimal results. On the one hand, entity resolution requires more than independent classification due to the interconnected nature of record relationships (Getoor and Machanavajjhala, 2012). For example, in record linkage (*i.e.,* clean-clean ER), a single record from one data source typically matches at most one record from another data source, since there are usually no duplicates in a single database. Unfortunately, matching-based approaches do not take advantage of this nature of record linkage. On the other hand, this strategy ignores the capabilities of LLMs to handle multiple records simultaneously to distinguish similar records. Using the records in Figure 1(c) as an example, if "Cruzer Glide", "Cruzer Force", and "Extreme Pro" appear in different records of the same context, LLMs are more likely to recognize that they are different San-

Disk flash drive models, which helps with accurate matching. As a result, the *matching* strategy cannot fully unleash the potential of LLMs in EM.

In this paper, we thoroughly investigate three strategies for LLM-based entity matching that incorporate record interactions from different perspectives, as shown in Figure 1. Specifically, apart from the conventional *matching* strategy shown in Figure 1(a), we investigate two additional strategies that leverage information from other records: 1) the *comparing* strategy, which identifies the record out of two that is more likely to match the anchor record, as shown in Figure 1(b); 2) the *selecting* strategy, which directly chooses the record from a list that is most likely to match the anchor record, as shown in Figure 1(c). Our research suggests that for LLM-based entity matching, incorporating record interactions is critical and can significantly improve EM performance in various scenarios. Therefore, the global *selecting* strategy is often the most effective. Nevertheless, we also observe that the selection accuracy decreases greatly as the position of the matching record increases in the candidate list. The position bias and limited long context understanding of current LLMs hinder the generality of the *selecting* strategy.

Based on our findings, we design a *compound entity matching framework* (ComEM) that leverages the composition of multiple strategies and LLMs. Specifically, given an entity record and its n potential matches obtained from the blocking step, we first preliminarily rank and filter these candidates using the local *matching* or *comparing* strategy, implemented with a medium-sized LLM. We then perform fine-grained identification on only the top k candidates using the global *selecting* strategy, facilitated by a more powerful LLM. This approach not only mitigates the challenges and biases faced by the *selecting* strategy with too many options, but also reduces the cost of LLM invocations caused by composing multiple strategies. Consequently, by integrating the advantages of different strategies and LLMs, ComEM achieves a more effective and efficient entity matching process.

To investigate different strategies and to evaluate our ComEM framework, we conducted in-depth experiments on 8 ER datasets. Experimental results verify the effectiveness of incorporating record interactions through the *selecting* strategy, with an average 17.58% improvement in F1 over the current *matching* strategy. In addition, we examined the effect of 9 different LLMs using these strategies on identification or ranking. Based on the results, ComEM is able to further improve the average F1 of the single *selecting* strategy by up to 4.01% while reducing the cost.

Contributions. Generally speaking, our contributions can be summarized as follows[1]:

- We investigate three strategies for LLM-based entity matching, and delve into their advantages and shortcomings in different scenarios.
- We design a ComEM framework by integrating the advantages of different strategies and LLMs to address the challenges of EM.
- We conduct thorough experiments to investigate these strategies for EM and verify the effectiveness of our proposed framework.

2 Related Work

2.1 Entity Resolution

As the core of data integration and cleaning, entity resolution has received extensive attention over the past decades (Fellegi and Sunter, 1969; Getoor and Machanavajjhala, 2012; Binette and Steorts, 2020; Papadakis et al., 2021). The blocking-and-matching pipeline has become the mainstream of entity resolution, where blocking filters out obviously dissimilar records and matching identifies duplicates through complex techniques.

Blocking. Traditional blocking approaches group records into blocks by shared signatures, followed by cleaning up unnecessary blocks and comparisons (Papadakis et al., 2022). Meta-blocking further reduces superfluous candidates by weighting potential record pairs and graph pruning (Papadakis et al., 2014). Recently, nearest-neighbor search techniques, especially cardinality-based ones, have gained more attention and achieved state-of-the-art (SOTA) results for blocking (Thirumuruganathan et al., 2021; Paulsen et al., 2023).

Entity Matching. The open and complex nature of entity matching has spurred the development of various approaches to address this persistent challenge, including rule-based (Benjelloun et al., 2009; Li et al., 2015), distance-based (Bilenko et al., 2003), and probabilistic methods (Fellegi and Sunter, 1969; Wu et al., 2020), etc. With the advent of deep learning methods (Mudgal et al., 2018), especially pre-trained language models (PLMs) (Li

[1]The source code of this paper is available at: anonymous.4open.science/r/LLM4EM and supplementary material.

et al., 2020), entity matching has made significant progress (Barlaug and Gulla, 2021; Tu et al., 2023). The emergence of LLMs brings a new zero- or few-shot paradigm to entity matching (Narayan et al., 2022; Peeters and Bizer, 2023a), alleviating training data requirements. Most deep learning and LLM-based approaches treat entity matching as an independent classification problem, except for GNEM (Chen et al., 2021), which models this task as a collective classification task on graphs. To the best of our knowledge, this is the first effort to formulate entity matching as a comparison or selection task using LLMs.

2.2 Large Language Model

The advent of LLMs such as ChatGPT marks a significant advance in artificial intelligence, offering unprecedented natural language understanding and generation capabilities, and even general intelligence (Bubeck et al., 2023). By scaling up the model and data size of PLMs, LLMs exhibit emergent abilities (Wei et al., 2022) and can thus solve a variety of complex tasks by "prompt engineering" without "fine-tuning". For more technical details on LLMs, we refer the reader to the related survey (Zhao et al., 2023).

3 Entity Matching with LLMs

In this section, we first present the problem formulation. Then, we introduce three strategies for LLM-based entity matching. Finally, we propose our ComEM framework, which leverages the composition of multiple strategies and LLMs.

3.1 Problem Formulation

We formulate the task of entity matching as the process of identifying matching records from a given entity record r and its n potential matches $R = \{r_1, r_2, \ldots, r_n\}$ obtained from blocking. This formulation mitigates the limitations of independent pairwise matching and fits real-world entity resolution scenarios. First, current SOTA blocking methods adhere to the k-nearest neighbor (kNN) search paradigm, which retrieves a list of potential matches for each entity record, rather than generating candidate matches pairwise as in traditional blocking workflows. In addition, this formulation accommodates both single-source deduplication and dual-source record linkage, and makes good use of the 1-1 assumption, *i.e.,* record r matches at most one of the potential matches R. This assumption is widespread in record linkage, and deduplication with canonical entity construction.

3.2 LLM as a Matcher

Recent work formulates entity matching as a binary classification task based on LLMs (Narayan et al., 2022; Peeters and Bizer, 2023b,a; Fan et al., 2023; Li et al., 2024). In this strategy, an LLM acts as a pairwise matcher to determine whether two records match. Specifically, given an entity record r and its potential matches $R = \{r_1, r_2, \ldots, r_n\}$, this approach independently classifies each pair of records $(r, r_i)_{1 \leq i \leq n}$ as matching or not by interfacing LLMs with an appropriate matching prompt, as shown in Figure 1(a):

$$\mathrm{LLM}_m\colon \{(r, r_i) \mid r_i \in R\} \rightarrow \{\mathrm{Yes}, \mathrm{No}\}$$

Unlike previous studies, the core of LLM-based applications is to prompt LLMs to generate the correct answer, namely prompt engineering. An appropriate prompt should include the task instruction, such as "*Do these two records refer to the same real-world entity? Answer Yes or No*". Optionally, a prompt could include detailed rules or several in-context learning examples to guide LLMs in performing this task. Given the need for long contexts in other strategies, and the instability of existing prompt engineering methods for entity matching (Peeters and Bizer, 2023a), we only attempt few-shot prompting for the matching strategy and leave the exploration of better prompt engineering with different strategies to future work.

This independent matching strategy ignores the global consistency of ER, as well as the capabilities of LLMs to incorporate record interactions. For record linkage, according to the well-known 1-1 assumption, each entity record r matches at most one record of the potential matches R. For deduplication, this assumption can also be satisfied by constructing canonical entities. The traditional solution to satisfy these constraints is to construct a graph based on the similarity scores s_i of record pairs (r, r_i) and to further cluster on the similarity graph. We can obtain the similarity scores from LLMs by calibrating the generated probabilities p of labels (Qin et al., 2023). Formally, the similarity score s_i can be defined as:

$$s_i = \begin{cases} 1 + p(\mathrm{Yes} \mid (r, r_i)), & \text{if generate "Yes"} \\ 1 - p(\mathrm{No} \mid (r, r_i)), & \text{if generate "No"} \end{cases}$$

Unfortunately, the generation probabilities are not available for many black-box commercial LLMs.

Moreover, the probabilities on short-form labels are misaligned for common open-source chat-tuned LLMs because they are fine-tuned to respond in detail. The need to investigate better strategies for LLM-based entity matching arises in ER.

3.3 LLM as a Comparator

In this section, we introduce a comparing strategy for LLM-based entity matching that simultaneously compares two potential matches to a given record. Specifically, given an entity record r and its potential matches $R = \{r_1, r_2, \ldots, r_n\}$, the comparing strategy compares two records r_i and r_j from potential matches R to determine which is more consistent with record r by interfacing LLMs with a comparison prompt, as shown in Figure 1(b):

$$\mathrm{LLM}_c\colon\ \{(r, r_i, r_j) \mid r_{i,j} \in R\} \to \{\mathrm{A}, \mathrm{B}\}$$

where A and B are labels corresponding to record r_i and r_j. Since LLMs may be sensitive to the prompt order, we compare the record pair (r_i, r_j) to record r twice by swapping their order.

Compared to the matching strategy, the comparing strategy introduces an additional record for more record interactions and shifts the task paradigm. It focuses on indicating the relative relationship between two potential matches of a given record, rather than making a direct match or no match decision. Therefore, this strategy is suitable for ranking and fine-grained filtering to determine the most likely records for identification.

To rank candidate records using the comparing strategy, we can compute similarity scores to estimate how closely each candidate matches the anchor record. Unlike the matching strategy, the comparing strategy can obtain similarity scores of record pairs using black-box LLMs, which do not provide probabilities. In such case, the similarity score s_i of record pair (r, r_i) can be defined as:

$$s_i = 2 \times \textstyle\sum_{j \neq i} \mathbb{1}_{r_i >_r r_j} + \sum_{j \neq i} \mathbb{1}_{r_i =_r r_j}$$

where $\mathbb{1}_{r_i >_r r_j}$ and $\mathbb{1}_{r_i =_r r_j}$ indicate that record r_i wins twice and once in comparison with record r_j to record r. When LLMs do provide probabilities, the similarity scores s_i can be defined as:

$$s_i = \textstyle\sum_{j \neq i} \left(p(\mathrm{A} \mid (r, r_i, r_j)) + p(\mathrm{B} \mid (r, r_j, r_i)) \right)$$

However, the advantage of the comparing strategy in obtaining similarity scores comes at the cost of using LLMs as the basic unit of comparison and $\mathcal{O}(n^2)$ complexity. Fortunately, for entity matching, we only care about a small number of most similar candidates, and there are many comparison sort algorithms available to find the top-k elements efficiently. In this paper, we use the *bubble sort* algorithm to find the top-k elements, optimizing the complexity of the comparing strategy to $\mathcal{O}(kn)$. To avoid confusion, we refer to the comparison of all pairs as $\texttt{comparing}_{\texttt{all-pair}}$ in our experiments.

3.4 LLM as a Selector

In this section, we introduce a selecting strategy that uses an LLM to select the matching record of a given record from a list of potential matches. Specifically, given an entity record r and its potential matches $R = \{r_1, r_2, \ldots, r_n\}$, this strategy directly selects the match of record r from R by interfacing LLMs with an appropriate selection prompt, as shown in Figure 1(c):

$$\mathrm{LLM}_s\colon\ \{(r, R)\} \to \{1, 2, \ldots, n\}$$

where $1, \ldots, n$ indicates the corresponding record.

In this way, LLMs can be explicitly required to identify only one match per record r from the potential matches R. Furthermore, feeding LLMs all potential matches in the same context at a time allows LLMs to make better decisions by considering interactions between records. For example, if "SanDisk Cruzer Glide", "SanDisk Cruzer Force", and "SanDisk Extreme Pro" appear in different records of the same context, it is easier for LLMs to recognize that these are different model names of SanDisk flash drives and select the actual match.

One challenge in applying the selecting strategy to LLM-based entity matching is that there is often no actual match of record r in potential matches R, because entity matching is an imbalanced task. A trivial solution to this challenge could be to perform a pairwise matching after the selection, which would undermine the advantages of the selecting strategy. Another method could be to add "none of the above" as an additional option to allow LLMs to refuse to select any record from the potential matches, which can be formulated as:

$$\mathrm{LLM}_{s_N}\colon\ \{(r, R)\} \to \{0, 1, 2, \ldots, n\}$$

where 0 indicates the "none of the above" option.

However, the selecting strategy relies heavily on the capabilities of LLMs for fine-grained understanding and implicit ranking in long contexts. Our experimental results show that the current

Strategy	Similarity Score	Interaction Level	# LLM Invocations	# Input Records
Matching	–	+	$\mathcal{O}(n)$	$2n$
Comparing	✓	++	$\mathcal{O}(kn)$	$3k(2n-k-1)$
Selecting	✗	+++	$\mathcal{O}(1)$	$n+1$

Table 1: Comparison of different strategies. "–" means that the matching strategy can only calibrate similarity scores if the generation probability is available. "# Input Records" represents the number of (#) records input to LLMs using different strategies for record r and its n potential matches R. k denotes the number of top candidates considered by the comparing strategy.

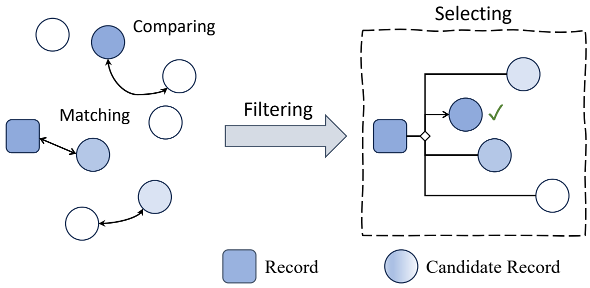

Figure 2: Illustration of COMEM framework. It first filters candidate records by matching or comparing strategies and then identifies the match via selecting strategy.

Dataset	Domain	# D1	# D2	# Attr	# Pos
Abt-Buy (AB)	Product	1076	1076	3	1076
Amazon-Google (AG)	Software	1354	3039	4	1103
DBLP-ACM (DA)	Citation	2616	2294	4	2224
DBLP-Scholar (DS)	Citation	2516	61353	4	2308
IMDB-TMDB (IM)	Movie	5118	6056	5	1968
IMDB-TVDB (IV)	Movie	5118	7810	4	1072
TMDB-TVDB (TT)	Movie	6056	7810	6	1095
Walmart-Amazon (WA)	Electronics	2554	22074	6	853

Table 2: Statistics of experimental datasets.

LLMs suffer from position bias, with the selection accuracy decreasing significantly as the position of the matching record increases in the candidate list (§ 4.3). In practice, the recall-oriented blocking step often generates a considerable number of potential matches for each record, exceeding the context length that LLMs can effectively reasoning (Levy et al., 2024). Therefore, it is a challenge to mitigate the position bias and the long context requirement for the selecting strategy.

3.5 Compound Entity Matching Framework

Based on the advantages and shortcomings of different strategies, we further propose a compound entity matching framework (COMEM). COMEM addresses various challenges in LLM-based entity matching by integrating the advantages of different strategies and LLMs. Table 1 shows a comparison of these strategies. The matching and comparing strategies are applicable for local ranking, while the selecting strategy is suitable for fine-grained identification. Therefore, as shown in Figure 2, we first utilize a medium-sized LLM to rank and filter potential matches R of record r with the matching or comparing strategy. We then utilize an LLM to identify the match of record r from only the top k candidates with the selecting strategy.

Our COMEM framework integrates the advantages of different strategies through a filtering then identifying pipeline. It first utilizes the local matching or comparing strategy to rank potential matches for preliminary screening, which can effectively mitigate the position bias and the long context requirement of the selecting strategy. It then utilizes the global selecting strategy to incorporate record interactions for fine-grained optimization, which can effectively mitigate the consistency ignorance of the matching strategy. Therefore, COMEM is able to strike a balance between entity matching requirements and current LLM capabilities, achieving significant performance improvements.

By integrating LLMs of different sizes, our COMEM framework can also effectively reduce the cost of LLM invocations for entity matching. In practice, direct use of commercial LLMs is expensive because entity matching is a computationally intensive task. COMEM delegates a significant part of the computation to medium-sized LLMs. Experimental results show that the ranking process can be performed well by using open-source medium-sized (3B~11B) LLMs (§ 4.4). As a result, the proper integration of LLMs not only improves the performance of entity matching but also reduces the cost for practical application.

4 Experiments

In this section, we conduct thorough experiments to investigate three strategies for LLM-based entity matching. First, we present the main experimental results (§ 4.2). Next, we perform the analysis of different strategies (§ 4.3). Finally, we examine the effect of different LLMs on these strategies (§ 4.4).

4.1 Experimental Setup

Datasets. We focused on the common record linkage that has many open-access datasets. Specifically, we used 8 clean-clean ER datasets collected by pyJedAI (Nikoletos et al., 2022). Table 2 shows the statistics of these datasets. We adapted them to fit the problem formulation and to support our

Method	Metric	AB	AG	DA	DS	IM	IV	TT	WA	Mean	Cost
Sudowoodo	P	71.43	38.00	86.71	71.19	84.44	57.81	67.97	73.49	68.88	**1.11**
	R	50.00	75.00	95.67	84.00	85.00	95.00	87.00	40.67	76.54	
	F1	58.82	50.45	**90.97**	77.06	84.72	71.88	76.32	52.36	70.32	
Matching	P	40.41	35.54	65.78	64.63	95.08	68.75	65.28	35.62	58.89	4.52
	R	91.33	59.00	98.67	81.00	58.00	55.00	94.00	88.33	78.17	
	F1	56.03	44.36	78.93	71.89	72.05	61.11	77.05	50.77	64.02	
Comparing	P	81.69	65.31	85.60	82.74	96.55	84.82	88.93	71.26	82.11	11.75
	R	77.33	42.67	69.33	54.33	46.67	31.67	85.67	60.33	58.50	
	F1	79.45	51.61	76.61	65.59	62.92	46.12	**87.27**	65.34	66.86	
Selecting	P	74.08	58.13	81.34	73.89	89.41	84.07	77.18	72.95	76.38	1.71
	R	87.67	70.33	97.33	88.67	95.67	82.67	91.33	89.00	87.83	
	F1	80.31	63.65	88.62	80.61	92.43	83.36	83.66	80.18	81.60	
ComEM	P	85.67	66.57	86.23	80.48	94.59	86.06	79.94	85.11	83.08	1.67
	R	89.67	73.00	96.00	89.33	99.00	82.33	90.33	87.67	88.42	
	F1	**87.62**	**69.63**	90.85	**84.68**	**96.74**	**84.16**	84.82	**86.37**	**85.61**	

Table 3: Overall performance and cost of different methods. We bold the **best** F1 score and underline the second best.

experiments. For each dataset with two record collections D1 and D2, we applied the SOTA blocking method Sparkly (Paulsen et al., 2023) to retrieve 10 potential matches from D2 for each record in D1. The recall@10 of Sparkly on all datasets ranges from 86.57% to 99.96%, demonstrating its effectiveness in retrieving potential matches. In this way, we are able to investigate and evaluate different strategies under the real ER pipeline.

Baseline. Except for the pairwise matching strategy, we also compare the STOA self-supervised learning method, Sudowoodo (Wang et al., 2023), which reduces the need for supervision through contrastive learning and pseudo-labeling.

Evaluation Metrics. Consistent with prior studies, we report **F1**, **P**recision, and **R**ecall on record pairs. We also report the cost ($) of LLM invocations[2].

Implementation Details.[3] We used ChatGPT (gpt-3.5-turbo-0613) as the main LLM for strategy analysis. We also examined the effect of 8 open-source *instruction-tuned* LLMs, including Llama-3-8B (AI@Meta, 2024), Qwen2-7B (Bai et al., 2023), Mistral-7B (Jiang et al., 2023), Mixtral-8x7B (Jiang et al., 2024), Flan-T5-XXL (Chung et al., 2022), Flan-UL2 (Tay et al., 2023) and Solar-10.7B (Kim et al., 2023). The specific prompts can be found in Appendix A, with the generation temperature of all LLMs set to 0 for reproducibility. For each dataset, we sampled 400 records from record collection D1 for evaluation, 300 of which had matches, and formed 4000 pairs of records with

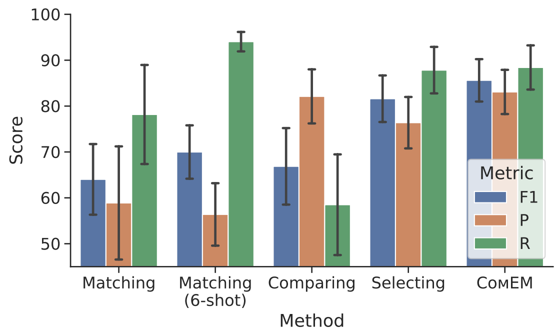

Figure 3: Comparison of different strategies.

their potential matches obtained by Sparkly from record collection D2. The remaining record pairs (unsampled records and their potential matches) are used for model training or in-context learning. For Sudowoodo, we used its official implementation[4] to train models on 500 pairs. For in-context learning, we select 100 record pairs and follow Peeters and Bizer (2023a) to retrieve 3 positives and 3 negatives as few-shot examples. Since the comparing strategy produces only relative orders, we applied the matching strategy to the top 1 candidate after bubble sort ranking. In ComEM, we used Flan-T5-XL to rank candidates with the matching strategy and kept the top 4 candidates for selection.

4.2 Main Results

We first compare the performance and cost of different methods, with the following findings.

Finding 1. *Incorporating record interactions is essential for LLM-based entity matching.* As shown in Table 3, entity matching performance in-

[2]The inference or training cost is estimated based on the hourly price of the cloud NVIDIA A40.

[3]We have provided the full code, including blocking and sampling in the Supplementary Material for reproducibility.

[4]https://github.com/megagonlabs/sudowoodo

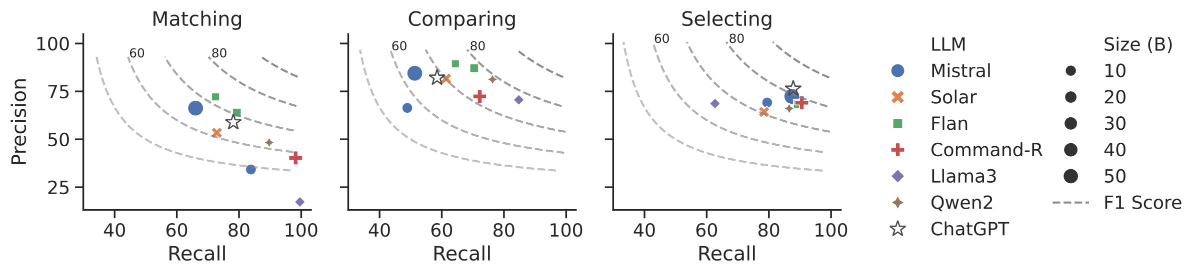

Figure 4: Effect of open-source LLMs on different strategies.

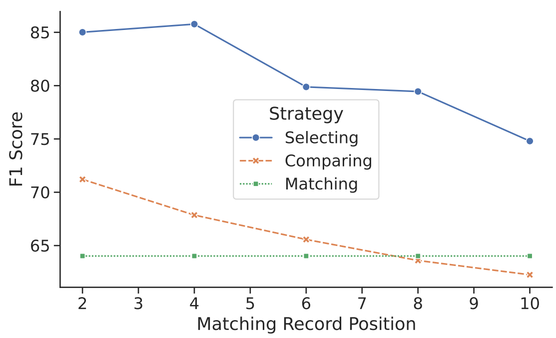

Figure 5: F1 score *w.r.t.* matched record positions.

creases as the level of record interaction increases. The comparing strategy outperforms the independent matching strategy by an average of 2.84% F1 score, and the selecting strategy further improves the F1 score by up to 14.74% over the comparing strategy. The optimal selecting strategy is 11.28% higher in F1 than the SOTA self-supervised learning method. The advantages of the comparing and selecting strategies over the matching strategy are also evident across different LLMs in Figure 4. To further verify that these improvements are due to the strategy, we perform 6-shot matching, ensuring that the number of records is consistent with the selecting strategy. As shown in Figure 3, the selecting strategy still has a significant F1 advantage over 6-shot matching. *These results highlight the effectiveness of our proposed strategies and open new avenues for LLM-based entity matching.*

Finding 2. *By integrating the advantages of different strategies and LLMs,* ComEM *can accomplish EM more effectively and cost-efficiently.* As shown in Table 3 and Figure 3, compared to the optimal selecting strategy using ChatGPT, ComEM achieves 4% F1 improvement while spending less. The filtering and identifying pipeline improves precision considerably (6.7%) without sacrificing high recall of the selecting strategy. These results reveal that integrating multiple strategies can complement single strategies and mitigate the position bias of the selecting strategy in long contexts. However, using a single powerful but costly commercial LLM to complete the entire pipeline obscures the cost efficiency of the selecting strategy. By introducing a medium-sized LLM for preliminary filtering, ComEM improves performance while spending less than direct selection. *As a result,* ComEM *underscores the importance of task decomposition and LLM composition, illuminating an effective route for compound entity matching using LLMs.*

4.3 Analysis of Strategies

We then analyze the advantages and shortcomings of different strategies from different perspectives.

Finding 3. *The selecting is the most cost-effective strategy for LLM-based entity matching.* Monetary cost is also an important factor when interfacing LLMs for EM in practice, as it is computationally intensive. As shown in Table 3, the selecting strategy costs less than half of the matching strategy. This is because the selecting strategy saves $n-1$ times of repeatedly inputting anchor records and task instructions into LLMs. The comparing strategy, however, considers two potential matches at a time and interfaces the LLM twice, making its cost more than twice that of the matching strategy. Therefore, the selecting strategy stands out for its effectiveness and efficiency.

Finding 4. *Strategies that incorporate multiple records suffer from the position bias of LLMs.* As shown in Figure 5, the performance of the comparing and selecting strategies decreases significantly as the position of the matching records moves down in the candidate list. For the comparing strategy optimized with bubble sort, matching records cannot be ranked at the top if there is any incorrect comparison. The selecting strategy also drops about 10% in F1, probably due to the limited long context understanding of the LLM. Therefore, the position bias of LLMs restricts the performance and gener-

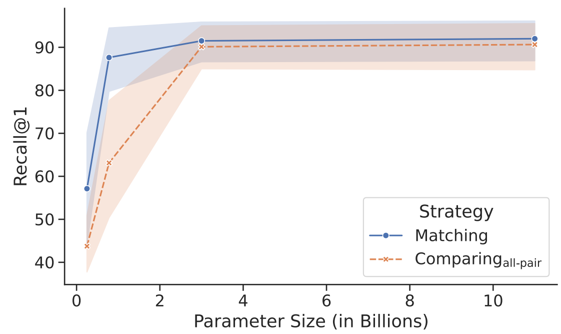

Figure 6: Ranking recall@1 *w.r.t.* model parameters.

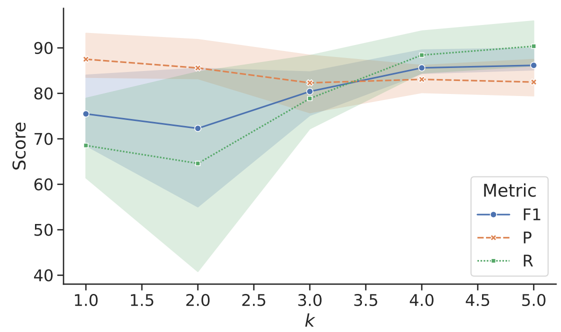

Figure 7: Average F1, precision, and recall *w.r.t.* number of candidate retained (k) for further selection. See Appendix C for detailed results on each dataset.

ality of the comparing and selecting strategies.

4.4 Effect of LLMs

We further examine the effect of open-source LLMs on these strategies to identify matches or rank.

Finding 5. *There is no single LLM that is uniformly dominant across all strategies.* Figure 4 shows the efficacy of proposed strategies for open-source LLMs, with detailed results in Appendix B. We can see that the F1 scores of the matching, comparing, and selecting strategies for different LLMs mostly fall between 50%~70%, 60%~80%, and 70%~80%, respectively. In general, similar to ChatGPT, the comparing strategy is better than the matching strategy, while the selecting strategy is further better than the comparing strategy. The consistent performance between strategies confirms the effectiveness of incorporating record interactions in these ways. Concretely, some chat LLMs such as Llama3-8B and Mistral-7B produce numerous false positives and thus perform poorly with the matching strategy. Nevertheless, they achieve significant improvement and satisfactory performance by comparing and selecting strategies, respectively. Moreover, although Flan-T5-XXL and Flan-UL2 lag behind ChatGPT by about 4% F1 in the selecting strategy, we find that they perform quite well in the matching and comparing strategies. These task-tuned LLMs follow instructions better than chat-tuned LLMs, and can output only the requested labels instead of long-form responses, making it convenient to utilize label generation probabilities. *In conclusion, there is a noticeable variance in the capabilities of different LLMs for a single strategy, and the efficacy of different strategies for a single LLM can also be significantly distinct.*

Finding 6. *Matching strategy is better for ranking and filtering than comparing strategy.* The superiority of Flan-T5 in the matching and comparing strategies leads us to explore the possibility of using it to rank and filter potential matches for the selecting strategy. As shown in Figure 6, the matching strategy outperforms the comparing strategy under different model parameter sizes, even though the latter performs $\mathcal{O}(n^2)$ comparisons. The difference is small on Flan-T5-XL (3B) and Flan-T5-XXL (11B), but significant on smaller models. This may be due to the fact that these models are trained on many pairwise tasks, such as natural language inference and question answering, but few triple-wise tasks. Therefore, in terms of effectiveness and efficiency, the matching strategy is more suitable for ranking and filtering potential matches.

4.5 Ablation Study

We perform an ablation study on the number of candidate records for further identification. As shown in Figure 7, recall increases and precision decreases as the number of retained potential matches increases. Consistent with Figure 5, four is the sweet spot for the selecting strategy with current ChatGPT, which balances precision and recall well.

5 Conclusion

In this paper, we investigate three strategies for entity matching using LLMs to bridge the gap between local matching and global consistency of ER. Our research shows that incorporating record interactions is essential for LLM-based entity matching. By examining the effect of broad LLMs on these strategies, we further design a ComEM framework that integrates the advantages of multiple strategies and LLMs. The effectiveness and cost efficiency of ComEM highlight the importance of task decomposition and LLM composition, opening up new avenues for entity matching using LLMs.

Limitations

This study aims to investigate different strategies for LLM-based entity matching. We conducted thorough experiments with 1 commercial LLM and 8 open-source LLMs to provide a broad base for our analysis. The selection of models is based on considerations of popularity, availability, and cost. Future research could explore whether similar findings hold as LLMs evolve and how performance changes relative to our results.

Since LLMs have been trained on massive web data, they are likely to have seen the similar and same records, or even some matching results, even though the labels of the matches are stored separately. *Nevertheless*, the performance of these strategies is relatively consistent across 9 LLMs and varies greatly for the same LLM when using different strategies, highlighting that data exposure is not the determining factor in their effectiveness. In the future, it will be valuable to evaluate LLM-based entity matching on new or non-public data.

The investigation of different strategies was conducted using basic zero/few-shot promting, a simple and effective paradigm for applying LLMs. We could not ignore the role of potential advanced prompt engineering methods in improving the accuracy and robustness of LLMs. In addition, fine-tuning LLMs for better execution of different strategies is also a worthwhile direction.

Finally, we have demonstrated the effectiveness of the compound framework in entity matching that integrates different strategies and LLMs. We would like to continue to develop specific modules for entity matching and extend this paradigm to different stages of entity resolution.

References

AI@Meta. 2024. Llama 3 model card.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Nils Barlaug and Jon Atle Gulla. 2021. Neural networks for entity matching: A survey. *ACM Trans. Knowl. Discov. Data*, 15(3):52:1–52:37.

Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *VLDB J.*, 18(1):255–276.

Mikhail Bilenko, Raymond J. Mooney, William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. Adaptive name matching in information integration. *IEEE Intell. Syst.*, 18(5):16–23.

Olivier Binette and Rebecca C. Steorts. 2020. (almost) all of entity resolution. *CoRR*, abs/2008.04443.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Runjin Chen, Yanyan Shen, and Dongxiang Zhang. 2021. GNEM: A generic one-to-set neural entity matching framework. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1686–1694. ACM / IW3C2.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16.

Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. 2023. Cost-effective in-context learning for entity resolution: A design space exploration. *CoRR*, abs/2312.03987.

Ivan P. Fellegi and Alan B. Sunter. 1969. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Lise Getoor and Ashwin Machanavajjhala. 2012. Entity resolution: Theory, practice & open challenges. *Proc. VLDB Endow.*, 5(12):2018–2019.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. SOLAR 10.7b: Scaling large language models with simple yet effective depth upscaling. *CoRR*, abs/2312.15166.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *CoRR*, abs/2402.14848.

Huahang Li, Longyu Feng, Shuangyin Li, Fei Hao, Chen Jason Zhang, Yuanfeng Song, and Lei Chen. 2024. On leveraging large language models for enhancing entity resolution. *CoRR*, abs/2401.03426.

Lingli Li, Jianzhong Li, and Hong Gao. 2015. Rule-based method for entity resolution. *IEEE Trans. Knowl. Data Eng.*, 27(1):250–263.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.*, 14(1):50–60.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 19–34. ACM.

Avanika Narayan, Ines Chami, Laurel J. Orr, and Christopher Ré. 2022. Can foundation models wrangle your data? *Proc. VLDB Endow.*, 16(4):738–746.

Konstantinos Nikoletos, George Papadakis, and Manolis Koubarakis. 2022. pyjedai: a lightsaber for link discovery. In *Proceedings of the ISWC 2022 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 21st International Semantic Web Conference (ISWC 2022), Virtual Conference, Hangzhou, China, October 23-27, 2022*, volume 3254 of *CEUR Workshop Proceedings*. CEUR-WS.org.

George Papadakis, Marco Fisichella, Franziska Schoger, George Mandilaras, Nikolaus Augsten, and Wolfgang Nejdl. 2022. How to reduce the search space of entity resolution: with blocking or nearest neighbor search? *CoRR*, abs/2202.12521.

George Papadakis, Ekaterini Ioannou, Emanouil Thanos, and Themis Palpanas. 2021. *The Four Generations of Entity Resolution*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

George Papadakis, Georgia Koutrika, Themis Palpanas, and Wolfgang Nejdl. 2014. Meta-blocking: Taking entity resolutionto the next level. *IEEE Trans. Knowl. Data Eng.*, 26(8):1946–1960.

Derek Paulsen, Yash Govind, and AnHai Doan. 2023. Sparkly: A simple yet surprisingly strong TF/IDF blocker for entity matching. *Proc. VLDB Endow.*, 16(6):1507–1519.

Ralph Peeters and Christian Bizer. 2023a. Entity matching using large language models. *CoRR*, abs/2310.11244.

Ralph Peeters and Christian Bizer. 2023b. Using chatgpt for entity matching. In *New Trends in Database and Information Systems - ADBIS 2023 Short Papers, Doctoral Consortium and Workshops: AIDMA, DOING, K-Gals, MADEISD, PeRS, Barcelona, Spain, September 4-7, 2023, Proceedings*, volume 1850 of *Communications in Computer and Information Science*, pages 221–230. Springer.

Ralph Peeters, Reng Chiz Der, and Christian Bizer. 2024. WDC products: A multi-dimensional entity matching benchmark. In *Proceedings 27th International Conference on Extending Database Technology, EDBT 2024, Paestum, Italy, March 25 - March 28*, pages 22–33. OpenProceedings.org.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large language models are effective text rankers with pairwise ranking prompting. *CoRR*, abs/2306.17563.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Saravanan Thirumuruganathan, Han Li, Nan Tang, Mourad Ouzzani, Yash Govind, Derek Paulsen, Glenn Fung, and AnHai Doan. 2021. Deep learning for blocking in entity matching: A design space exploration. *Proc. VLDB Endow.*, 14(11):2459–2472.

Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Guoliang Li, Xiaoyong Du, Xiaofeng Jia, and Song Gao. 2023. Unicorn: A unified multi-tasking model for supporting matching tasks in data integration. *Proc. ACM Manag. Data*, 1(1):84:1–84:26.

Runhui Wang, Yuliang Li, and Jin Wang. 2023. Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 1502–1515. IEEE.

Tianshu Wang, Hongyu Lin, Cheng Fu, Xianpei Han, Le Sun, Feiyu Xiong, Hui Chen, Minlong Lu, and Xiuwen Zhu. 2022. Bridging the gap between reality and ideality of entity matching: A revisting and benchmark re-constrcution. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3978–3984. ijcai.org.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Renzhi Wu, Sanya Chaba, Saurabh Sawlani, Xu Chu, and Saravanan Thirumuruganathan. 2020. Zeroer: Entity resolution using zero labeled examples. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1149–1164. ACM.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Matching
Do the two entity records refer to the same real-world entity? Answer "Yes" if they do and "No" if they do not. Record 1: {{ record_left }} Record 2: {{ record_right }}
Comparing
Which of the following two records is more likely to refer to the same real-world entity as the given record? Answer with the corresponding record identifier "Record A" or "Record B". Given entity record: {{ anchor }} Record A: {{ candidate_left }} Record B: {{ candidate_right }}
Selecting
Select a record from the following candidates that refers to the same real-world entity as the given record. Answer with the corresponding record number surrounded by "[]" or "[0]" if there is none. Given entity record: {{ anchor }} Candidate records:{% for candidate in candidates %} [{{ loop.index }}] {{ candidate }}{% endfor %}

Table 4: Specific prompts of different strategies. We use JinJa template syntax to display the placeholders for the *anchor* record and potential matches (*candidates*).

A Prompts

The prompts for various strategies of LLM-based entity matching used in this paper are presented in Table 4. To ensure fairness, the same prompts were used for all experimental LLMs. These prompts were constructed through a manual process of prompt engineering, which involved the testing and comparing of multiple variations to determine the most effective ones. In addition to the task description, we included specific response instructions such as "*Answer "Yes" if they do and "No" if they do not*" to guide the responses of LLMs. For in-context learning, prompts and labels were repeatedly inputted for each example, followed by the records to be matched. We post-processed the LLM responses to obtain the final predicted labels.

B Detailed Results of Open-Source LLMs under Different Strategies

We provide the detailed F1 scores of open-source LLMs under different strategies in Table 5. Among the 8 LLMs evaluated in our experiment, 6 achieve the best performance through the selecting strategy, and 2 achieve better performance through the comparing strategy. In summary, our proposed strategies are universally applicable across different LLMs for entity matching. We have observed that it is difficult to limit the output of many chat-tuned LLMs simply by prompts, which may affect their actual performance in entity matching. Therefore, how to calibrate the label probabilities from the long-form responses of LLMs is also important for performance improvement.

C Ablation Study on Each Dataset

Figure 8 shows the performance of ComEM with respect to the varying number of top candidates retained for further selection. Similar to Figure 7, recall increases and precision decreases as k increases. For the simplest dataset "DBLP-ACM", F1 achieves the highest value at $k = 1$. For some other datasets, F1 changes dramatically as k goes from 2 to 5. How to tune k to balance the LLM capabilities and the actual situation is a direction that could be explored. One possible solution might be to dynamically adjust the number according to the similarity scores of potential matches.

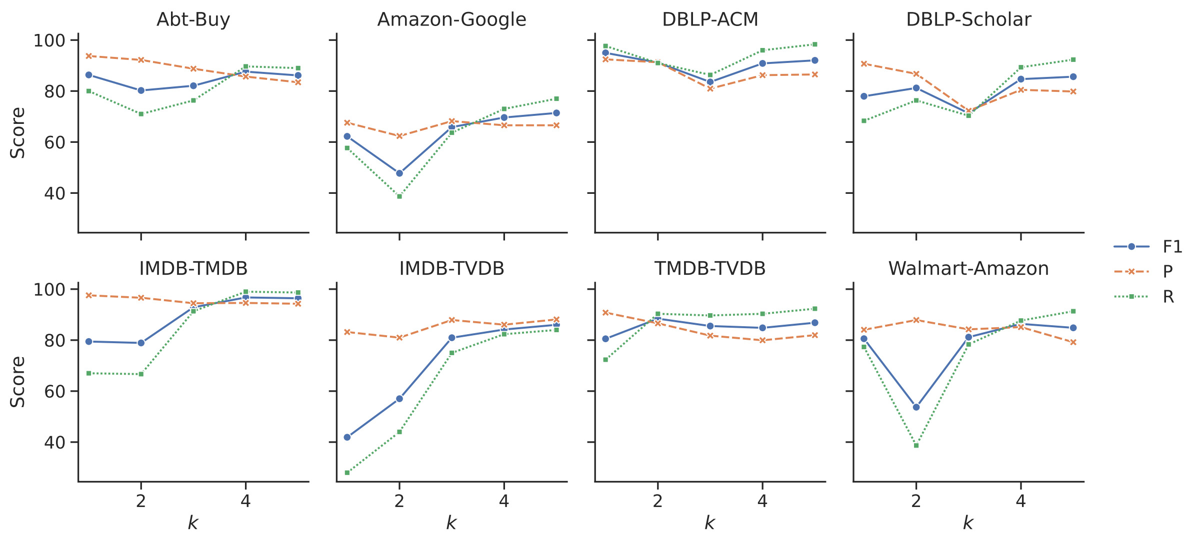

Figure 8: F1, precision, and recall *w.r.t.* the number of candidates retained (k) for further selection on all datasets.

LLM	Strategy	AB	AG	DA	DS	IM	IV	TT	WA	Mean
Mistral-Instruct-7B	Matching	40.70	37.77	24.68	28.89	64.86	64.49	49.91	55.96	45.91
	Comparing	54.68	32.10	49.28	49.75	74.38	52.25	81.69	44.39	54.82
	Selecting	67.26	57.31	83.36	74.27	87.84	76.95	80.89	62.54	73.80
Mistral-Instruct-8x7B	Matching	77.67	34.76	67.20	60.09	82.26	53.57	72.99	50.57	62.39
	Comparing	67.81	25.20	81.48	75.54	75.15	54.05	73.93	41.22	61.80
	Selecting	79.58	61.16	85.05	79.37	90.34	77.15	81.23	78.84	79.09
Solar-Instruct-10.7B	Matching	68.80	45.60	47.02	38.32	70.35	40.49	75.18	70.57	57.04
	Comparing	86.22	49.14	84.70	75.16	61.68	32.57	77.49	74.41	67.67
	Selecting	74.27	62.05	74.93	65.50	79.56	59.68	73.96	74.89	70.60
Flan-T5-XXL (11B)	Matching	77.85	58.35	87.63	80.34	71.82	51.62	74.62	67.23	71.18
	Comparing	84.21	56.85	94.49	85.82	65.33	49.88	84.28	67.89	73.60
	Selecting	77.52	69.83	84.77	80.29	85.07	68.05	78.90	77.33	77.72
Flan-UL2 (20B)	Matching	83.39	52.73	81.97	67.53	82.35	40.56	70.88	74.07	69.19
	Comparing	88.09	64.52	94.81	88.26	71.43	39.51	83.66	80.66	76.37
	Selecting	80.34	71.82	84.00	80.57	84.09	65.70	80.99	71.94	77.43
Command-R-35B	Matching	49.87	32.87	47.87	44.46	91.45	69.69	63.14	36.81	54.52
	Comparing	72.31	51.27	76.82	65.91	90.91	77.00	86.09	57.24	72.20
	Selecting	78.16	65.52	83.67	79.54	85.26	75.33	79.06	80.58	78.39
Llama-3-8B	Matching	31.01	21.97	19.27	19.27	44.78	40.24	31.55	23.91	29.00
	Comparing	80.06	61.27	84.85	72.54	80.13	76.36	79.82	80.29	76.91
	Selecting	74.37	49.50	78.91	68.79	76.27	54.77	69.66	42.33	64.33
Qwen-2-7B	Matching	63.41	47.33	68.35	52.46	82.89	55.54	71.84	55.06	62.11
	Comparing	84.32	56.88	88.78	76.57	93.17	65.07	86.50	75.39	78.34
	Selecting	72.39	61.03	81.49	76.57	82.97	73.48	78.55	72.96	74.93

Table 5: F1 score of open-source LLMs under different strategies.