Entity Matching with 7B LLMs: A Study on Prompting Strategies and Hardware Limitations

Ioannis Arvanitis-Kasinikos[1], George Papadakis[1]

[1]National and Kapodistrian University of Athens, Greece

Abstract

Entity Matching (EM) is a fundamental task in data management, involving the identification and linking of records that refer to the same real-world entity across different datasets. While Large Language Models (LLMs) have shown promise in addressing complex natural language processing tasks, their substantial computational requirements often limit their practical applicability. In this work, we investigate the use of 7B parameter LLMs with 4-bit quantization for EM tasks executable on commodity hardware. We explore various prompting strategies, including zero-shot, few-shot, and general matching definition prompts, to evaluate their effectiveness in improving EM accuracy. Experiments are conducted on two benchmark datasets with products, which present varying levels of complexity and challenge in product descriptions. Our findings demonstrate that 7B parameter LLMs can effectively perform EM, with the Orca2 model consistently outperforming others across different prompting strategies and datasets. The study highlights that few-shot prompting significantly enhances performance over zero-shot approaches, emphasizing the importance of task-specific examples and careful prompt design. We also examine the impact of example order in few-shot prompts and find that it has a substantial effect on model performance. Finally, we examine hardware limitations, demonstrating that effective EM can be achieved with resource-constrained models.

Keywords

Entity Matching, 7B LLMs, Zero-Shot Prompts, Few-Shot Prompts

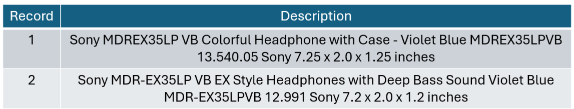
1. Introduction

Entity Resolution (ER) constitutes a vital task in data management that involves identifying and linking records from different datasets that refer to the same real-world entity [1, 2]. In many domains, including e-commerce, healthcare, and finance, accurate ER is essential for ensuring data quality, enabling effective data integration, and supporting informed decision-making [3]. However, this task is challenging due to data inconsistencies, incompleteness, and ambiguity across different sources [4, 5].

As an example, consider the product descriptions in Figure 1. Despite corresponding to the same object (Sony headphones), there are significant variations in product names, attributes, and dimensions. These discrepancies illustrate the challenges in reconciling variations across datasets, particularly when dealing with unstructured text and linguistic differences. Accurate ER in scenarios like this is crucial for product catalog integration, price comparison, and recommendation systems [6].

Due to its quadratic time complexity, ER solutions typically implement the Filtering-Verification framework [7]. The Filtering step, often called Blocking, significantly reduces the computational cost to the most similar candidate pairs, which are the most likely matches [8]. The Verification step performs Entity Matching (EM), which essentially determines whether two records are duplicates, describing the same real-world object. In the following, we exclusively focus on EM.

Traditional EM solutions typically rely on rule-based approaches, string similarity metrics, or machine learning algorithms [9, 10, 11]. However, these methods can struggle with complex linguistic variations and contextual understanding, while requiring domain expertise and heavy human involvement [12]. This is addressed by more recent state-of-the-art approaches that leverage deep learning (DL) techniques [13]. However, they require substantial amounts of training data, which are rarely available.



Record	Description
1	Sony MDREX35LP VB Colorful Headphone with Case - Violet Blue MDREX35LPVB 13.540.05 Sony 7.25 x 2.0 x 1.25 inches
2	Sony MDR-EX35LP VB EX Style Headphones with Deep Bass Sound Violet Blue MDR-EX35LPVB 12.991 Sony 7.2 x 2.0 x 1.2 inches

Figure 1: Two records with major differences describing the same product.

Recent advancements in NLP, particularly in Large Language Models (LLMs), offer new possibilities for addressing EM challenges [14, 15]. LLMs possess advanced capabilities for natural language understanding, which allows them to process and interpret complex textual descriptions [16]. Most importantly, LLM-based EM can be performed in zero-shot settings, requiring no training instances, a characteristic particularly attractive for out-of-the-box solutions.

In this work, we evaluate the performance of 7B parameter LLMs in entity matching tasks. While larger LLMs with hundreds of billions of parameters have shown impressive results [15, 16], their computational requirements often make them impractical for many real-world applications. By employing these LLMs, which excel in natural language understanding and semantic similarity assessment, this work seeks to address EM challenges in real-world datasets with linguistic variations and unstructured text, while also highlighting their suitability for execution on commodity hardware. The focus on 7B parameter LLMs is motivated by their potential for efficient deployment on commodity hardware, making them more suitable for practical applications.

To this end, we perform an extensive experimental evaluation that considers the models' ability to handle different types of EM scenarios. We explore *novel* zero-shot, few-shot, and general matching definition prompting strategies to assess their effectiveness in improving matching accuracy. Our goal is to bridge the gap between the advanced capabilities of LLMs and the practical constraints of real-world EM applications, potentially paving the way for more efficient and accurate ER techniques in diverse domains.

DOLAP 2025: 27th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data, co-located with EDBT/ICDT 2025, March 25, 2025, Barcelona, Spain

cs1180001@di.uoa.gr (I. Arvanitis-Kasinikos); gpapadis@di.uoa.gr (G. Papadakis)
https://gpapadis.wordpress.com (G. Papadakis)
0000-0002-7298-9431 (G. Papadakis)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

2. Related Work

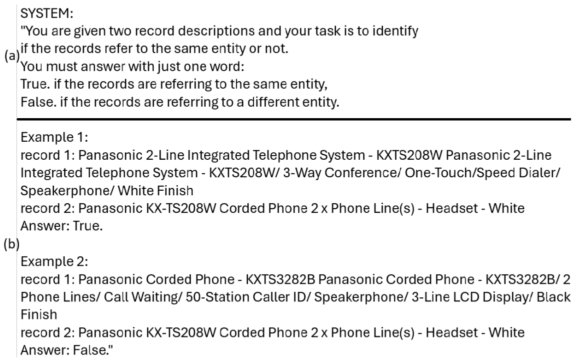
There is a plethora of recent LLM-based EM methods, because LLMs offer several advantages over traditional EM solutions: (i) contextual understanding, as they understand the context and semantics of entity descriptions better than traditional string matching techniques. (ii) robustness, since LLMs are typically more capable of addressing variations in how entity information is expressed. (iii) zero-shot and few-shot learning, i.e., LLMs can accomplish EM tasks with no or minimal examples of matching decisions. These characteristics render LLMs ideal for most EM tasks, especially those with complex, unstructured product descriptions.

The seminal work on LLM-based EM [16] investigated the effectiveness of GPT3-175B in EM, focusing on three key parameters: (i) problem definition, exploring different phrasings such as "Are Product A and Product B the same?" or "Are Product A and Product B equivalent?". (ii) in-context learning, comparing zero-shot with few-shot approaches. The former involve prompts with no examples in the prompt, while the latter involve a couple of examples, which are selected randomly or by experts. (iii) entity serialization, testing the use of all attributes or just a subset of them. Their experimental analysis led to the following conclusions: (i) few-shot learning significantly outperforms zero-shot approaches, (ii) attribute selection yields better results than using all attributes, (iii) problem definition has a substantial impact on performance, (iv) LLM performance is comparable to the state-of-the-art DL-based matching algorithms.

A detailed study was conducted in [15], using six LLMs, three hosted and three open-source ones. The experiments explored additional parameters such as problem definition, language complexity, output specification, entity serialization, in-context learning, instructions, and fine-tuning. The experimental results revealed that: (i) no single prompt consistently outperformed all others across different scenarios. (ii) Open-source LLMs showed comparable effectiveness to hosted models. (iii) LLMs performed competitively with deep learning-based matchers, even in zero-shot settings.(iv) Few-shot and instruction-based prompts generally outperformed zero-shot approaches. (v) Fine-tuning significantly improved effectiveness.

In another line of research, three distinct prompting strategies were explored in [17]: (i) Match prompts, which contain traditional pair-wise questions. E.g., "Do these two records refer to the same real-world entity? Record 1: [details]. Record 2: [details]." (ii) Comparison prompts, which ask for the most similar entity to a given reference. E.g., "Which of these two records is more consistent with the given record? Given Record: [details]. (A) Record 1: [details]. (B) Record 2: [details]." (iii) Selection prompts, which identify a matching entity from a set of candidates. E.g., "Select a record from the following list that refers to the same real-world entity as the given record: Given Record: [details]. Options: 1. [details] 2. [details] 3. [details]..." The experimental results show that incorporating record interactions through the comparison and selection prompts significantly improves EM performance across various scenarios; among the two, the selection prompts are the top-performers in most cases. However, they suffer from position bias, because their accuracy decreases when the duplicate record is placed lower in the list of candidates.

BatchER [18] aims to reduce the costs for hosted LLMs through batch processing, exploring various methods for question batching and demonstration selection. The experimental results demonstrate that batch prompting outperform match prompts in both effectiveness and cost, with the top performance achieved by diversity-based question batching combined with covering-based demonstration selection.



```
SYSTEM:
"You are given two record descriptions and your task is to identify
if the records refer to the same entity or not.
You must answer with just one word:
True. if the records are referring to the same entity,
False. if the records are referring to a different entity.

Example 1:
record 1: Panasonic 2-Line Integrated Telephone System - KXTS208W Panasonic 2-Line
Integrated Telephone System - KXTS208W/ 3-Way Conference/ One-Touch/Speed Dialer/
Speakerphone/ White Finish
record 2: Panasonic KX-TS208W Corded Phone 2 x Phone Line(s) - Headset - White
Answer: True.

Example 2:
record 1: Panasonic Corded Phone - KXTS3282B Panasonic Corded Phone - KXTS3282B/ 2
Phone Lines/ Call Waiting/ 50-Station Caller ID/ Speakerphone/ 3-Line LCD Display/ Black
Finish
record 2: Panasonic KX-TS208W Corded Phone 2 x Phone Line(s) - Headset - White
Answer: False."
```

Figure 2: (a) The basic zero-shot EM prompt, and (b) its few-shot extension.

These studies collectively demonstrate the potential of LLMs in entity matching tasks, highlighting the importance of prompt engineering, the competitiveness of open-source models, and the effectiveness of batching strategies for improved efficiency. This work builds upon and extends the existing ones by focusing specifically on 7B parameter LLMs with 4-bit quantization. Unlike previous studies that primarily use larger, more resource-intensive models, our work explores the potential of smaller and more accessible LLMs for EM tasks. In this context, we perform a comprehensive evaluation of various *novel* prompting strategies, including zero-shot, few-shot, and general matching definition approaches, across multiple models and datasets. This approach offers insights into the practical applicability of LLMs in resource-constrained environments, bridging the gap between advanced language models and real-world EM challenges.

3. Problem Definition

Applied after Filtering, Entity Matching is typically formulated as a binary classification problem [3, 4]. More formally: *Given two records r_1 and r_2, the task is to determine whether they refer to the same entity.* This is often expressed as a function $f(r_1, r_2) \rightarrow \{0, 1\}$, where 1 indicates a match (also called duplicate) and 0 indicates a non-match.

In LLM-based settings, EM is framed as a natural language inference task. The LLM is provided with descriptions of two records and asked to determine if they refer to the same entity, returning "True" for a match and "False" otherwise.

In all cases, EM performance is measured with respect to:

- Precision, i.e., the proportion of correctly identified matches out of all predicted matches.
- Recall, i.e., the proportion of correctly identified matches out of all actual matches.
- F-measure, i.e., the harmonic mean of precision and recall, providing a balanced measure of performance.
- Run-time, i.e., the time taken to complete the ER process.

The first three measures are defined in $[0, 1]$ with higher values indicating higher *effectiveness.* For the last one, lower values indicate higher *time efficiency.*

4. EM Prompts

We now present the EM prompts that are examined in our work. The basic prompt is presented in Figure 2(a). It consists of an instruction that describes the input and the desired output. It lacks any examples, thus constitutes a *zero-shot EM prompt*, which tests the model's ability to generalize to new tasks or domains it has not been trained on.

A concise *few-shot EM prompt* extends the zero-shot one with the examples in Figure 2(b). To provide a balanced context, there are two examples that include a pair of matching entities and a pair of non-matching ones. These examples serve as a form of weak supervision, allowing the LLM to learn from the provided instances and generalize to similar cases. Note that the examples in Figure 2(b) have been carefully selected from dataset D_1 (see Table 1) so that they capture typical variations in product descriptions that are encountered in the full dataset.

Note that LLM responses to few-shot prompts suffer from *position bias* [17], because the order of examples in the EM prompt might alter the matching decision. This means that in the example of Figure 2(b), the response for a specific candidate pair might be True (i.e., matching) if the positive example precedes the negative one and False (i.e., non-matching) otherwise. For this reason, we define two types of few-shot prompts:

1. *TF*, where the True example is followed by False one, as in Figure 2(b).

2. *FT*, where the False example is followed by True one.

Note that with multiple examples per prompt, as in [17], more arrangements are possible. In this work, though, we exclusively consider the two variations of the few-shot EM prompt that involves one example per match type.

To increase the robustness of LLMs to few-shot EM prompts, we consider two matching approaches for each candidate pair, query with both TF and FT prompts:

1. The *union approach* labels a candidate pair as True if either the TF or FT prompt results in a True response.

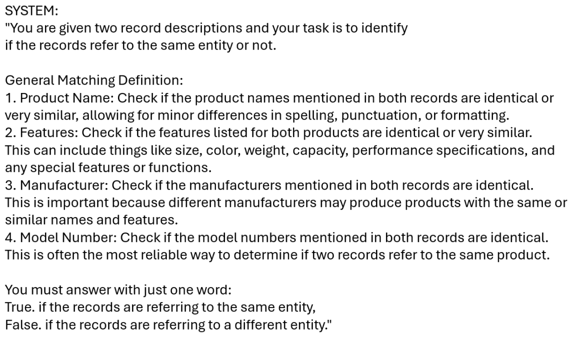
2. The *intersection approach* labels a candidate pair as True only if both the TF and FT prompts yield a True response.

4.1. Domain-specific Zero-Shot Prompts

The above prompts are generic enough to apply to any domain. In our experimental analysis, we also consider *domain-specific* ones, which are crafted for the product matching task. More specifically, we devise a zero-shot prompt that involves general matching definitions, providing the LLM with explicit guidance on how to determine if two records refer to the same product.

The core assumption of this approach is that the records are described by a clean, aligned schema. This is necessary for building a schema-aware generic definition of duplicate records. In the product matching task, we use four key product attributes: (i) product name, (ii) features, (iii) manufacturer, and (iv) model number. We use them in two different configurations:

1. The *composite domain-specific EM prompt* concatenates all four criteria in the above sequence, as in Figure 3. The goal is to facilitate more nuanced matching decisions.



```
SYSTEM:
"You are given two record descriptions and your task is to identify
if the records refer to the same entity or not.

General Matching Definition:
1. Product Name: Check if the product names mentioned in both records are identical or
very similar, allowing for minor differences in spelling, punctuation, or formatting.
2. Features: Check if the features listed for both products are identical or very similar.
This can include things like size, color, weight, capacity, performance specifications, and
any special features or functions.
3. Manufacturer: Check if the manufacturers mentioned in both records are identical.
This is important because different manufacturers may produce products with the same or
similar names and features.
4. Model Number: Check if the model numbers mentioned in both records are identical.
This is often the most reliable way to determine if two records refer to the same product.

You must answer with just one word:
True. if the records are referring to the same entity,
False. if the records are referring to a different entity."
```

Figure 3: Domain-specific, zero-shot EM prompt for product matching.

2. The *atomic domain-specific EM prompt* uses only the model number as the matching criterion. We selected this attribute because it provides the cleanest and most distinctive values.

These two configurations were chosen after preliminary tests that suggested that they yield the best performance among all other combinations of these four attributes.

5. Experimental Analysis

Experimental Settings. All experiments were implemented in Python v3.12.0 and Ollama[1] v0.1.22. All experiments were carried out on a server running Ubuntu 22.04.1 LTS, equipped with Intel Core i7-9700K 8 core @ 3.6 GHz, 32GB RAM and NVIDIA GeForce GTX 1080 Ti 11GB.

Due to the limited size of the available VRAM, our study focuses on 7-billion-parameter LLMs with optimizations such as *quantization*, which in our case replaces the 32-bit floating-point model weights with 4-bit integers. This reduces the model size, while maintaining reasonable performance levels. In other words, quantization lowers effectiveness, due to the fewer parameters and the lower precision of the model's weights, but significantly reduces run-times and memory consumption. Therefore, our experimental results are useful for resource-constrained applications, which run LLMs on commodity hardware.

LLMs. There is a plethora of open-source LLMs, with newer models introduced on a rather frequent basis. During our study, two models were quite popular: Llama 2 [19], with 7B parameters and a context length of 4096, as well as Mistal [20], with 7.3B parameters. However, preliminary experiments demonstrated that both of them were inappropriate for the EM tasks considered in this work. Llama 2 consistently responded with "True" for every candidate pair, while Mistral failed to provide a response according to given instructions – it indicated an inability to respond in certain cases or gave explanations for its decisions instead of a "True" or "False" label.

In their place, we considered the following open-source models, which demonstrated high effectiveness in our preliminary experiments:

1. *Orca2* [21]. Built by Microsoft Research, Orca2 is a family of models fine-tuned on Meta's Llama 2 using synthetic data.

2. *OpenHermes*[2]. This is a Mistral 7B model fine-tuned with fully open datasets, showcasing strong multi-turn chat

[1] https://ollama.com
[2] https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B

Dataset	#Entities	Duplicates	Cartesian Product	#Attributes	Candidate Pairs	Bl.Recall	Bl.Precision
D_1	1,076-1,076	1,076	1.16×10^6	3	4,345	0.924	0.229
D_2	2,554-22,074	853	5.64×10^7	6	5,163	0.910	0.150

Table 1
Technical characteristics of the datasets used in the experimental analysis.

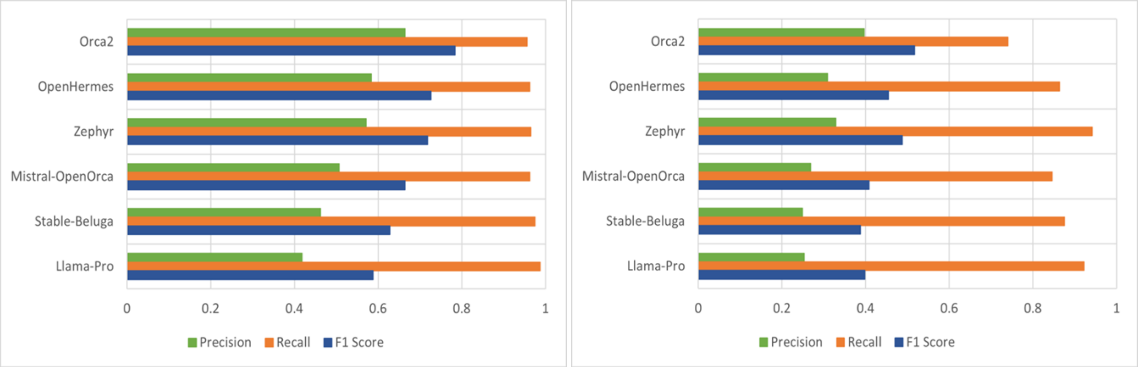


Figure 4: Effectiveness of the zero-shot prompt in Figure 2(a) on top of the selected LLMs over D_1 (left) and D_2 (right).

skills and system prompt capabilities. It surpasses all previous versions of Nous-Hermes 13B and below.

3. *Zephyr* [22]. A 7B parameter model fine-tuned on Mistral, it achieves results similar to Llama 2 70B Chat in various benchmarks. It is trained on a distilled dataset, improving grammar and chat results.

4. *Mistral-OpenOrca*[3]. This is a 7B parameter model, fine-tuned on top of Mistral 7B using the OpenOrca dataset.

5. *Stable-Beluga*[4]. This is a Llama 2 based model fine-tuned on an Orca-style dataset.

6. *Llama-Pro* [23]: An 8B parameter expansion of Llama 2 that specializes in integrating both general language understanding and domain-specific knowledge, particularly in programming and mathematics.

In all cases, we use the default latest model with 4-bit quantization and 7B parameters.

Datasets. We used two real-world datasets with products that are widely used in the ER literature: (i) D_1 is the Abt-Buy dataset, which comprises product listings from two online retailers, Abt Electronics and Buy.com. (ii) D_2 is the Walmart-Amazon dataset, which contains product listings from two other online retailers, Walmart and Amazon. D_1 primarily focuses on electronic products, while D_2 covers a broader range of product categories, matching diverse entity types. Both datasets present important challenges, such variations in product names and descriptions across retailers, inconsistent use of model numbers and other identifiers, differences in the level of detail provided for each product, variations in formatting and units (e.g., dimensions, weights) as well as missing or null values in certain fields.

Their technical characteristics are summarized in Table 1. Note that each dataset comprises two individually clean data sources, whose sizes are reported in column #Entities. Note also that we apply the prompts to the candidate pairs generated by a state-of-the-art blocking implemented by PyJedAI [24] , version 0.1.6. Following [25], we kNN-Join, which identifies the k nearest neighbors of each entity. We fine-tuned it, maximizing blocking precision for a blocking recall of at least 90%, as reported in the rightmost columns of Table 1. This configuration uses cleaning (i.e., stop-word removal and stemming) and cosine similarity in both datasets. For Abt-Buy, k was set to 4, while the attribute values were converted into a multiset of character trigrams. For Walmart-Amazon, k was set to 2, while the attribute values were converted into a multiset of character four-grams.

5.1. Zero-Shot Prompting Results

We now examine the relative performance of the selected LLMs over D_1 and D_2, when coupled with the basic zero-shot EM prompt of Figure 2(a).

We observe that Orca2, OpenHermes, and Zephyr consistently rank as the top three models with respect to F-Measure in both datasets. The last two models switch their ranking positions in the two datasets, whereas Orca2 maintains the lead. The superior performance of Orca2, which demonstrates Orca2's robustness under diverse EM settings, can be attributed to its fine-tuning on synthetic data designed for reasoning tasks. This enhances its capability to understand and compare complex product descriptions. OpenHermes is fine-tuned on fully open datasets with strong multi-turn chat skills, leveraging advanced language understanding to perform well. Zephyr's competitive performance probably results from its training on a distilled dataset that improves grammar and chat results, aiding in better interpretation of entity attributes. The lower performance of Mistral-OpenOrca, Stable-Beluga, and Llama-Pro is probably due to the less specialized training data or the smaller model capacities for the specific nuances of EM.

Note that all models exhibit much higher recall than precision in both datasets. This means that they are prone to label a candidate pair as matching, at the cost of introducing numerous false positives. Orca2 consistently exhibits the highest precision, thus yielding the highest F-Measure, too.

Note also that all models exhibit markedly lower effectiveness in D_2 compared to D_1. This suggests that D_2 presents

[3] https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca
[4] https://huggingface.co/stabilityai/StableBeluga2

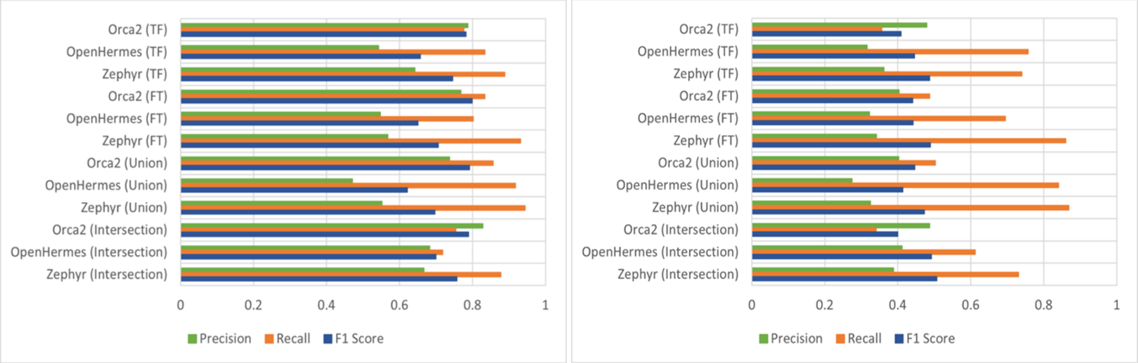


Figure 5: Effectiveness of the few-shot prompts in Figure 2(b) on top of selected LLMs over D_1 (left) and D_2 (right). From top to bottom, the TF promps are presented first, followed by the FT prompts, the Union and the Intersection approaches.

greater EM challenges, potentially due to more diverse or complex product descriptions. While D_1 is restricted to electronics, D_2 covers a broader range of products and includes more variation in descriptions, attributes, and data quality, rendering EM more difficult. Furthermore, D_1 has a 1:1 matching between its two data sources, whereas D_2 has a much lower ratio of matches, adding another layer of complexity to the task. The substantial performance gap between D_1 and D_2 underscores the significant impact of data characteristics on model effectiveness.

5.2. Few-Shot Prompting Results

We now examine the performance of the aforementioned few-shot prompts over D_1 and D_2. We disregard Mistral-OpenOrca, Stable-Beluga, and Llama-Pro, because they exhibited significantly lower effectiveness and less consistent performance in the zero-shot experiments – preliminary experiments verified their poor performance in few-shot settings, too. For brevity, we focus on the top three performing models, namely Orca2, OpenHermes, and Zephyr.

The results are reported in Figure 5. Based on preliminary experiments, we randomly select the examples included in the few-shot prompts from the candidate pairs of the same dataset. The same examples are used in all prompts issued on a particular dataset.

In both datasets, we observe the same patterns as regards the relative performance of TF and FT few-shot prompts: For Orca2, there is a substantial improvement when using the latter; OpenHermes is more robust to position bias, as there is no significant difference between the two prompt strategies; Zephyr works best when coupled with the TF few-shot prompts. These patterns highlight that the impact of position bias on each model is consistent across the two datasets. Note also that with the exception of Orca2 with TF prompts, all models achieve higher recall than precision, remaining more prone to label a candidate pair as matching.

It is also interesting to compare the union approach with the intersection one. For OpenHermes and Zephyr, the latter yields significantly higher F-Measure: by considering as duplicates only the candidate pairs that are marked as matching by both TF and FT few-shot prompts, the reduction in recall is much lower than the increase in precision (as a result, recall remains much higher than precision for both models). This means that considering only the common matches of TF and FT prompts leads to more accurate performance. Note that these patterns are consistent for both models over both datasets.

This is not the case with Orca2, whose performance varies significantly across the two datasets. In D_1, the same F1 score is achieved for both approaches, because the intersection raises recall by 12%, while reducing precision to the same degree. In D_2, though, the intersection reduces recall by 23% and increases precision by 16%, thus yielding a much lower F-Measure. Note that in both datasets, the recall of the model gets lower than its precision in combination with the intersection approach, unlike the union one.

Overall, we can conclude that Orca2 works best when coupled with FT few-shot prompts, while OpenHermes and Zephyr maximize their effectiveness when intersecting the matches of TF and FT prompts. Among them, the top performers over D_1 and D_2 are Orca2 (F1=0.799) and Zephyr (F1=0.531), respectively.

5.3. Domain-specific Zero-Shot Prompting Results

In this section, we compare the atomic domain-specific prompt with the composite one. As in Section 5.2, we exclusively consider the three top performing models with respect to the zero-shot prompts: Orca2, OpenHermes, and Zephyr. Their performance is reported in Figure 6.

We observe that in all cases, the atomic prompt outperforms the composite one to a significant extent – the only exception corresponds to Zephyr in D_1, where the composite prompt increases F-Measure almost by 15%. This pattern should be attributed to the short, distinctive and clean values provided by the model number. This way, it reduces the noise from other product attributes like product name, which are typically associated with long and diverse texts.

Similar to the above strategies, all LLMs exhibit much higher recall than precision. This means that they remain prone to mark a candidate pair as a match at the cost of introducing false positives – a behavior that permeates all prompt strategies we have examined.

Among the three models, Orca2 is consistently better, albeit to a minor extent in D_2. This consistent performance underscores Orca2's effectiveness in EM tasks under quite different prompt designs.

We can conclude that domain-specific zero-shot prompts offer an effective and reliable alternative in datasets with a clean schema of known characteristics.

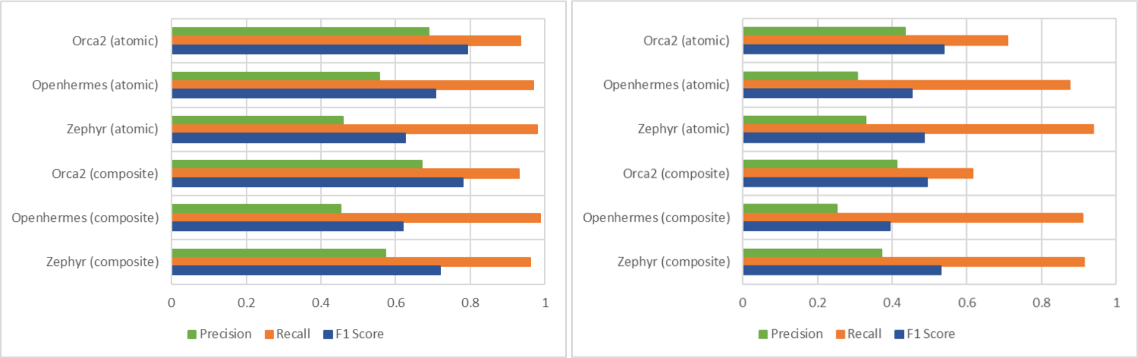


Figure 6: Effectiveness of the atomic and composite domain-specific zero-shot prompts in Figure 2(a) on top of the selected LLMs over D_1 (left) and D_2 (right).

Prompt Strategy	D_1				D_2			
	Precision	Recall	F-Measure	Run-time	Precision	Recall	F-Measure	Run-time
Zero-shot	0.664	0.956	0.784	32 min	0.397	0.740	0.517	23 min
FT Few-shot	0.768	0.834	**0.799**	41 min	0.420	0.515	0.463	33 min
Atomic Domain-specific	0.689	0.934	0.793	33 min	0.434	0.708	**0.538**	25 min

(a) Orca2

Prompt Strategy	Precision	Recall	F-Measure	Run-time	Precision	Recall	F-Measure	Run-time
Zero-shot	0.584	0.963	0.727	31 min	0.309	0.864	0.455	23 min
Intersection Few-shot	0.683	0.718	0.700	40 min	0.378	0.585	0.459	33 min
Atomic Domain-specific	0.556	0.969	0.707	33 min	0.306	0.876	0.453	25 min

(b) OpenHermes

Prompt Strategy	Precision	Recall	F-Measure	Run-time	Precision	Recall	F-Measure	Run-time
Zero-shot	0.572	0.965	0.718	32 min	0.329	0.942	0.488	24 min
Intersection Few-shot	0.667	0.877	0.757	43 min	0.408	0.761	0.531	34 min
Composite Domain-specific	0.573	0.960	0.718	39 min	0.372	0.913	0.529	30 min

(c) Zephyr

Table 2
Best performance per LLM in combination with the top performing variant per prompt strategy across both datasets.

5.4. Comparison of Prompting Strategies

We now compare the three top-performing models (Orca2, OpenHermes, and Zephyr) with respect to effectiveness and time efficiency across the three strategies of EM prompts discussed in Section 4. Note that among the few-shot and domain-specific variants, for each LLM we only consider the one with the highest F-Measure in both datasets. Their performance is reported in Table 2.

For Orca2, we observe that the FT few-shot prompts are the top performers in D_1. The atomic domain-specific ones follow in very close distance in terms of F-Measure, while exhibiting a much lower run-time. This means that the domain-specific prompts offer a significantly better balance between effectiveness and time efficiency. In D_2, this strategy scores the highest F-Measure for a slightly higher run-time than the second best approach (zero-shot prompts). For these reasons, Orca2 works best in combination with the atomic domain-specific prompts.

Regarding OpenHermes, the differences between the three types of prompts are minor in terms of F-Measure. As expected, the fastest approach in both datasets corresponds to the zero-shot prompts. This configuration also achieves the highest F-Measure in D_1, while in D_2, it ranks second, within a negligible distance from the top (<0.5%). Therefore, we can conclude that the zero-shot prompts are the best choice for OpenHermes.

For Zephyr, there is a clear winner in the case of D_1: the intersection of few-shot prompts. It exhibits, though, the highest run-time by a large extent. This is expected, as it queries the LLM twice per candidate pair. In the case of D_2, the same strategy takes a minor lead over the composite domain-specific prompts, which are faster by more than 10%. Due to its consistency, the best choice for Zephyr corresponds to the intersection of TF and FT few-shot prompts.

Method	D_1	Source	D_2	Source
ZeroER	0.520	[26]	0.644	[27]
Magellan	0.436	[28]	0.719	[28]
DeepMatcher	0.628	[28]	0.669	[28]

Table 3
The F-Measure per dataset reported in the literature for three state-of-the-art EM algorithms.

Among the three 7B LLMs, the configuration consistently achieving (almost) the highest effectiveness in both datasets is Orca2 coupled with atomic domain-specific prompts. Its efficiency is also rather high, given that its run-time is marginally higher than that of the fastest (zero-shot) configuration of the other two models.

5.5. Comparison to Baselines

To put the performance of the selected 7B LLMs into perspective, we compare it with three state-of-the-art EM approaches from the literature:

1. ZeroER [26], an unsupervised approach that requires no labelled datasets, learning Gaussian mixture models for matching and non-matching candidate pairs.

2. Magellan [29], a supervised approach combining binary classifiers with a series of hand-crafted features based on string similarity measures.

3. DeepMatcher [28], a framework leveraging the synergy between language models and Deep Learning classification.

For each method, we consider its best performance as reported in the literature. The results are reported in Table 1.

We observe mixed patterns. In D_1, all LLM configurations in Table 2, even the zero-shot prompts, outperform all three baseline methods to a significant extent ($> 21\%$). This is remarkable, because the simplest prompt strategy requires neither domain expertise nor the labeling candidate pairs, unlike Magellan and DeepMatcher, whose performance is derived from large training and validation sets, which amount to 60% and 20% of all candidate pairs, resp.

The situation is reversed in D_2, where all baseline methods achieve a much better performance. In fact, the highest F-measure of Orca2 is lower by 16.5% than the worst baseline (ZeroER). This should be attributed to the more challenging settings of D_2, which have already been discussed in Section 5.1. Note also that the records in D_2 are noisier, with a much higher portion of missing values. Its records are also longer, an aspect that is crucial for the 7B LLMs we are considering in this study, due to their limited attention window. These settings favor the learning-based functionality of the baseline methods, which take a clear lead over the learning-free functionality of 7B LLMs. Another reason for the poor performance of the latter is that they emphasize recall at the expense of precision, significantly decreasing their F-Measure in D_2, due the very low portion of matches in comparison to the total number of entities from each data source. Therefore, more advanced strategies are required for boosting the performance of 7B LLMs in datasets with characteristics similar to that of D_2.

6. Conclusions & Future Work

Focusing on 7B open-source LLMs, we examined the performance of three main prompt strategies: (i) the basic, domain-agnostic zero-shot prompt, (ii) the few-shot prompt with one example per type of matches, and (iii) the domain-specific zero-shot prompt. We considered several variants for the last two strategies and applied all of them on two established benchmark datasets for product matching. Testing six popular LLMs, we reached the following conclusions:

- Few-shot and domain-specific prompting significantly improve the performance of the zero-shot approaches, highlighting the value of task-specific prompts.

- In few-shot prompts, the response of LLMs is generally sensitive to order of examples. This suggests that a careful prompt engineering is crucial for optimal performance in real-world ER applications.

- This sensitivity can be addressed by the intersection approach to few-shot prompting, which consistently achieves much better results, increasing precision at a higher rate than it reduces recall.

- Orca2 consistently outperformed the other LLMs across most prompting strategies and datasets, demonstrating high robustness and effectiveness. In fact, the relative performance of the best models (Orca2 > OpenHermes > Zephyr) remained largely consistent across prompt strategies and datasets, suggesting inherent strengths in their base architectures.

- The use of 4-bit quantization and 7B parameter models demonstrated the potential for effective EM with limited computational resources. The effectiveness of the considered models is competitive with established, learning-based EM approaches, especially in datasets with low portion of missing values and short entity descriptions.

In the future, we plan to explore LLMs' capability in matching entities across different languages and to enhance the interpretability and explainability of LLM decisions.

Acknowledgments. This work was partially funded by the EU project STELAR (Horizon Europe – Grant No. 101070122).

References

[1] P. Christen, Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer, 2012.

[2] G. Papadakis, E. Ioannou, E. Thanos, T. Palpanas, The Four Generations of Entity Resolution, Morgan & Claypool Publishers, 2021.

[3] X. L. Dong, D. Srivastava, Big data integration, in: ICDE, 2013, pp. 1245–1248.

[4] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, K. Stefanidis, An overview of end-to-end entity resolution for big data, ACM Comput. Surv. 53 (2021) 127:1–127:42.

[5] K. Stefanidis, V. Efthymiou, M. Herschel, V. Christophides, Entity resolution in the web of data, in: 23rd International World Wide Web Conference, WWW, 2014, pp. 203–204.

[6] X. L. Dong, Building a broad knowledge graph for products, in: ICDE, 2019, p. 25.

[7] P. Christen, A survey of indexing techniques for scalable record linkage and deduplication, IEEE Trans. Knowl. Data Eng. 24 (2012) 1537–1555.

[8] G. Papadakis, D. Skoutas, E. Thanos, T. Palpanas, A survey of blocking and filtering techniques for entity resolution, CoRR abs/1905.06167 (2019).

[9] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, Duplicate record detection: A survey, IEEE Trans. Knowl. Data Eng. 19 (2007) 1–16.

[10] A. Jurek, J. Hong, Y. Chi, W. Liu, A novel ensemble learning approach to unsupervised record linkage, Inf. Syst. 71 (2017) 40–54.

[11] P. Christen, Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: SIGKDD, 2008, pp. 151–159.

[12] J. Fisher, P. Christen, Q. Wang, Active learning based entity resolution using markov logic, in: PAKDD, 2016, pp. 338–349.

[13] G. Papadakis, E. Ioannou, T. Palpanas, Entity resolution: Past, present and yet-to-come, in: EDBT, 2020, pp. 647–650.

[14] K. Nikoletos, E. Ioannou, G. Papadakis, The five generations of entity resolution on web data, in: ICWE, 2024, pp. 469–473.

[15] R. Peeters, C. Bizer, Entity matching using large language models, CoRR abs/2310.11244 (2023).

[16] A. Narayan, I. Chami, L. J. Orr, C. Ré, Can foundation models wrangle your data?, Proc. VLDB Endow. 16 (2022) 738–746.

[17] T. Wang, H. Lin, X. Chen, X. Han, H. Wang, Z. Zeng,

L. Sun, Match, compare, or select? an investigation of large language models for entity matching, CoRR abs/2405.16884 (2024).

[18] M. Fan, X. Han, J. Fan, C. Chai, N. Tang, G. Li, X. Du, Cost-effective in-context learning for entity resolution: A design space exploration, CoRR abs/2312.03987 (2023).

[19] H. Touvron, et al., Llama 2: Open foundation and fine-tuned chat models, CoRR abs/2307.09288 (2023).

[20] A. Q. Jiang, A. Sablayrolles, A. Mensch, et al., Mistral 7b, CoRR abs/2310.06825 (2023).

[21] A. Mitra, L. D. Corro, S. Mahajan, et al., Orca 2: Teaching small language models how to reason, CoRR abs/2311.11045 (2023).

[22] L. Tunstall, E. Beeching, N. Lambert, et al., Zephyr: Direct distillation of LM alignment, CoRR abs/2310.16944 (2023).

[23] C. Wu, Y. Gan, Y. Ge, Z. Lu, J. Wang, Y. Feng, Y. Shan, P. Luo, Llama pro: Progressive llama with block expansion, in: ACL, 2024, pp. 6518–6537.

[24] K. Nikoletos, G. Papadakis, M. Koubarakis, pyjedai: a lightsaber for link discovery, in: ISWC Posters, Demos and Industry Tracks, volume 3254, 2022.

[25] F. Neuhof, M. Fisichella, G. Papadakis, K. Nikoletos, N. Augsten, W. Nejdl, M. Koubarakis, Open benchmark for filtering techniques in entity resolution, VLDB J. 33 (2024) 1671–1696.

[26] R. Wu, S. Chaba, S. Sawlani, X. Chu, S. Thirumuruganathan, Zeroer: Entity resolution using zero labeled examples, in: SIGMOD, 2020, pp. 1149–1164.

[27] G. Papadakis, N. Kirielle, P. Christen, T. Palpanas, A critical re-evaluation of record linkage benchmarks for learning-based matching algorithms, in: ICDE, 2024, pp. 3435–3448.

[28] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, Deep learning for entity matching: A design space exploration, in: SIGMOD, 2018, pp. 19–34.

[29] P. Konda, S. Das, et al., Magellan: Toward building entity matching management systems, Proc. VLDB Endow. 9 (2016) 1197–1208.