BoostER: Leveraging Large Language Models for Enhancing Entity Resolution

Huahang Li
Hong Kong Polytechnic University
Hong Kong
hua-hang.li@connect.polyu.hk

Shuangyin Li
South China Normal University
Guangzhou, China
shuangyinli@scnu.edu.cn

Fei Hao
Hong Kong Polytechnic University
Hong Kong
ffaye.hao@polyu.edu.hk

Chen Jason Zhang*
Hong Kong Polytechnic University
Hong Kong
jason-c.zhang@polyu.edu.hk

Yuanfeng Song
Webank Co. Ltd.
Shenzhen, China
yfsong@webank.com

Lei Chen
Hong Kong University of Science and Technology
Hong Kong
leichen@cse.ust.hk

ABSTRACT

Entity resolution, which involves identifying and merging records that refer to the same real-world entity, is a crucial task in areas like Web data integration. This importance is underscored by the presence of numerous duplicated and multi-version data resources on the Web. However, achieving high-quality entity resolution typically demands significant effort. The advent of Large Language Models (LLMs) like GPT-4 has demonstrated advanced linguistic capabilities, which can be a new paradigm for this task. In this paper, we propose a demonstration system named **BoostER** that examines the possibility of leveraging LLMs in the entity resolution process, revealing advantages in both easy deployment and low cost. Our approach optimally selects a set of matching questions and poses them to LLMs for verification, then refines the distribution of entity resolution results with the response of LLMs. This offers promising prospects to achieve a high-quality entity resolution result for real-world applications, especially to individuals or small companies without the need for extensive model training or significant financial investment.

CCS CONCEPTS

• **Information systems** → **Entity resolution**; **Language models**.

KEYWORDS

Entity Resolution, Web Data Integration, Large Language Models

ACM Reference Format:
Huahang Li, Shuangyin Li, Fei Hao, Chen Jason Zhang, Yuanfeng Song, and Lei Chen. 2024. BoostER: Leveraging Large Language Models for Enhancing Entity Resolution. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651245

*Chen Jason Zhang is the corresponding author.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651245

1 INTRODUCTION

Living in the ear of the Web, we are surrounded by a vast array of information. Yet, the Web information is often messy, duplicated, or represented in myriad forms but refers to the same entity. Table 1 exhibits a Web application acting as an online repository for specialists, such as LinkedIn, but contains duplicated records in its database. Entity resolution focuses on identifying and merging the duplicated or associated records and creating a complete and precise representation of each entity. This process can utilize a range of techniques, such as deterministic matching, which depends on exact attribute matches, and probabilistic or machine learning-based methods. These approaches take into account multiple aspects and utilize statistical models to calculate the matching probabilities [4, 11]. A typical entity resolution workflow includes several crucial steps for ensuring accurate and reliable outcomes, which involve data preprocessing, record blocking, pairwise comparison, scoring and thresholding, and eventually clustering [2]. The outcome of entity resolution is a refined database possessing consolidated and unique representations of distinct real entities. This standard workflow is the basis for entity resolution tasks and has been commonly deployed in diverse areas to address the challenge of discovering and combining replicated data records.

However, most of the previous efforts focused on constructing entity resolution tools have approached the challenge as a classification problem. Within this framework, classifiers are carefully constructed to efficiently and precisely distinguish between pairs that are duplicates and those that are distinct, thereby categorizing these pairs based on similarity [1]. As a result, achieving optimal performance typically requires the design and training of specific models tailored to particular datasets. This approach restricts the model's applicability to limited scenarios and presents difficulties in adapting them to different domains. Though there are initiatives to develop a more generalized entity resolution model, the performance of these broader models remains imperfect [9].

A practicable method for enhancing entity resolution results involves integration with external knowledge sources, such as verification by crowd workers or insights from LLMs [8, 10]. In recent years, we have witnessed significant advancements in LLMs like GPT-4, Claude2, etc.. These LLMs, trained on vast and diverse datasets, excel at capturing intricate linguistic patterns, contextual nuances, and semantic meanings [6, 12]. An impressive strength

Table 1: Database Contents: Profiles of Professionals in the Web Application. Ground Truth: Records r_1 and r_2 correspond to the same individual, as do records r_3 and r_4.

ID	Name	Email	Title	Company	Loc.
r_1	John Doe	johndoe @email.com	Software Engineer	TechCorp	SF
r_2	J. Doe	johndoe @email.com	Software Engineer	TechCorp LLC	SF, CA
r_3	Jane Smith	janesmith @email.com	Project Manager	Innovate Tech	New York
r_4	Jane S.	janesmith @email.com	PM	Innovate Tech	NY
r_5	Johnathan Doe	johnathan.d @email.com	Developer	TechCorp	SF

of LLMs is their capability in contextual understanding and disambiguation, which have proven useful in handling ambiguous references and inconsistencies in entity attributes—elements that have traditionally been challenging in this domain. This endows them with a remarkable capability to match and compare entity attributes across varied records, enabling more precise and thorough outcomes. However, most LLMs, such as OpenAI's GPT-4, are charged for online API requests based on the total number of tokens in both the input (prompt or question) and the output (model's response)[1]. For example, if an API request includes an input of 10 tokens and yields an output of 20 tokens, the billing will be for 30 tokens in total. The token, which is the smallest unit processed by the model, can vary in size, representing anything from individual words to subwords or characters. Given that posing all matching questions could result in substantial costs, efficiently and effectively leveraging LLMs for enhancing entity resolution has turn into a crucial challenge.

In this paper, we introduce a cost-effective demonstration system named **BoostER**, which makes full use of the power of LLMs as a service. In our theoretical approach, every possible partition of the entity resolution results is taken into account to ensure the result set encompasses any conceivable scenario. Each possible partition is assigned with a probability that reveals the likelihood of it being accurate, as shown in Table 2. In practical applications, partitions are generated by basic entity resolution tools. We aggregate all the entity resolution results and then normalize these probabilities to sum up to 1. Subsequently, the probability of each potential matching pair is calculated by summing up the probabilities across its possible partitions. As depicted in Figure 1, each record is denoted by a node in a graph, and each link associated with a probability is represented as the potential matching pair. Therefore, the objective of entity resolution can be viewed as identifying potential linkages between these nodes. We adopt Shannon entropy to gauge the uncertainty of possible partitions. The underlying principle is that reducing entropy in a fixed system requires an external energy source, which is precisely the role LLMs fulfill. Our tailored greedy algorithm selects an optimal set of matching questions within the given budget, balancing the effectiveness of the matching questions against the cost of the number of tokens. Subsequently, we request these matching questions to LLMs for verification. Based on the responses from LLMs, we refine the probability distribution of possible partitions and recalculate the probabilities of possible matches. After several iterations of adjustments or upon exhausting the budget, a precise distribution of the entity resolution results is achieved.

[1]https://openai.com/pricing

Table 2: Probability Distribution of Possible Partitions in Table 1. The sum of all these probabilities equals to 1.

Possible Partition	Probability
$P_1 = \{(r_1, r_2), (r_3, r_4), (r_5)\}$	0.5
$P_2 = \{(r_1, r_2, r_3), (r_4, r_5)\}$	0.3
$P_3 = \{(r_1, r_3), (r_2, r_4), (r_5)\}$	0.2

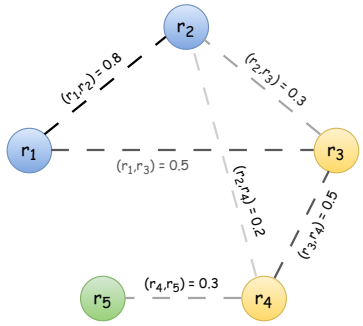


Figure 1: An illustration of Possible Matches (linkages) in Table 2. The Probability of each linkage is the cumulative sum of its occurrences across Possible Partitions.

2 THE BOOSTER FRAMEWORK

In this section, we present a detailed description of the BoostER framework. As depicted in Figure 2, BoostER encompasses several key steps: (1) Initialization of the Probability Distribution, (2) Selection of an Optimal Set of Matching Questions, (3) Verification by LLMs and Refinement of the Probability Distribution based on the responses. The technical specifics of each step are elaborated in the subsequent parts of this section.

2.1 Probability Distribution Initialization

The BoostER workflow begins by inputting records with duplicates, which may originate from multiple databases. Upon receiving these records, BoostER initially employs existing entity resolution tools to create a set of possible matches, each accompanied by a corresponding probability. An appropriate threshold is resorted to filter pairs with low probabilities. Then, to initiate the probability distribution of the possible partitions, we treat each pair as independent of the others, regarded as the Bernoulli distribution. Subsequently, we utilize Shannon entropy to assess the uncertainty of the results. A higher entropy value indicates greater incongruity among various entity resolution tools, signifying high uncertainty of the results.

2.2 Matching Questions Selection

We quantified the occurrence of matching pairs across all possible partitions to determine the probability of each pair. This probability, derived from statistical analysis, indicates the likelihood of a

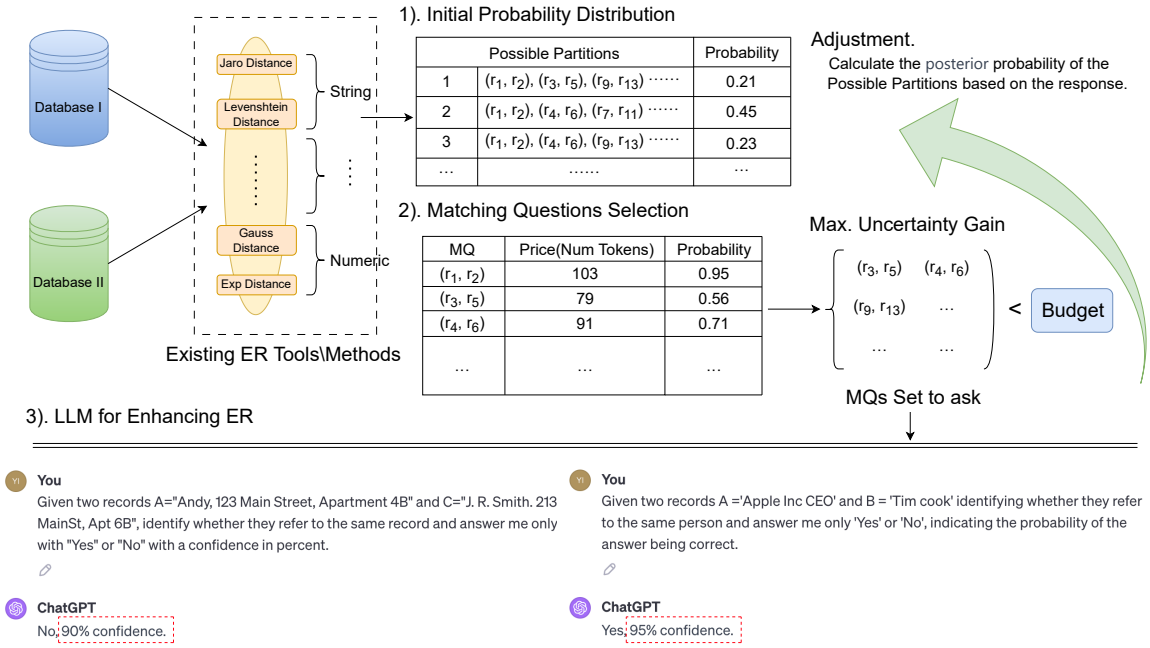


Figure 2: The Workflow of BoostER.

potential linkage between two records. A probability closer to 1 or 0 suggests higher certainty about the matching pair, whereas a value near 0.5 indicates greater uncertainty. Intuitively, one might select questions with the most uncertainty, for example, those with probabilities near 0.5. However, the selection process is complicated by the correlation between matching pairs. For instance, if we already get "$r_1 = r_2$" and "$r_2 = r_3$", then querying whether "$r_1 = r_3$" is unnecessary due to the transitivity property.

To find an optimal set of matching questions that leads to the most uncertainty reduction of possible distribution, we have established that the reduction in uncertainty is equivalent to the entropy of the answer set of these questions, denoted as D_A, and irrespective of the answering capability of LLMs. For a detailed proof, please refer to [7]. Thus, our problem turns out to be maximizing the joint entropy of the answer set within a given budget. Considering the cost associated with each question, we have devised a price function to convert the question to a constant price based on OpenAI's Tokenizer. Following these steps, we can effectively calculate both the anticipated benefits and the associated costs for any given set of matching questions.

The selection problem can be easily proven to be NP-hard. However, given that the joint entropy is a sub-modular function, our tailored greedy algorithm is capable of achieving near-optimal results. Specifically, it guarantees an approximation ratio of $(1 - 1/e)$. This constitutes the main contribution of our work.

2.3 Adjustment with LLMs Response

In real applications, even with the most advanced LLM, errors can still occur. The capability of an LLM, denoted as Θ, can be defined by the expected accuracy rate from the answers generated in specific tasks. This can be estimated by performing sample questions before starting. Our BoostER framework is designed to be error-tolerant. Below, we provide a running example demonstrating the adjustment process with LLMs response.

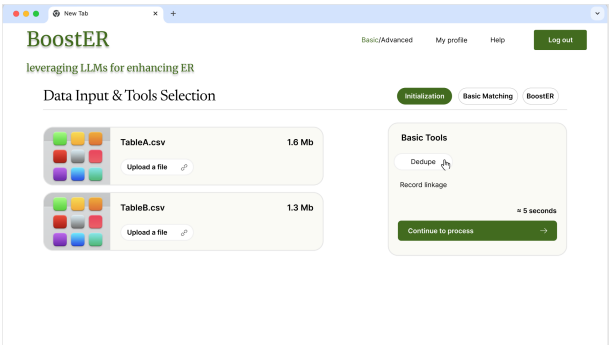
Running Example: Proceeding with the example in Figure 1, assume the record pair (r_3, r_4) is recognized as correct by the LLM and the LLM's capability is 90%, we can deduce:

$$\begin{aligned} &\mathcal{P}(P_1|Ans : (r_3, r_4) \textit{ is answered from LLM}) \\ =&\frac{\mathcal{P}(P_1)\mathcal{P}(Ans|P_1)}{\mathcal{P}(Ans)} \\ =&\frac{\mathcal{P}(P_1)\mathcal{P}(\Theta)}{\mathcal{P}(r_3, r_4)\mathcal{P}(\Theta) + (1 - \mathcal{P}(r_3, r_4)\mathcal{P}(1 - \Theta)} \\ =&\frac{0.5 * 0.9}{0.5 * 0.9 + 0.5 * 0.1} = 0.9, \end{aligned} \tag{1}$$

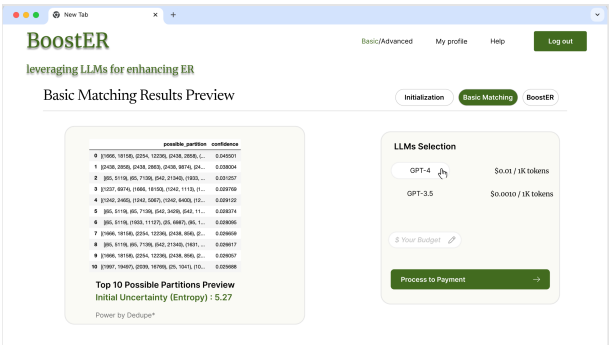
where $\mathcal{P}(\cdot)$ is the probability function. Similarly, for P_2, we have

$$\begin{aligned} &\mathcal{P}(P_2|Ans : (r_3, r_4) \textit{ is answered from LLM})) \\ =&\frac{\mathcal{P}(P_2)\mathcal{P}(Ans|P_2)}{\mathcal{P}(Ans)} \\ =&\frac{\mathcal{P}(P_2)\mathcal{P}(\Theta)}{\mathcal{P}(r_3, r_4)\mathcal{P}(1 - \Theta) + (1 - \mathcal{P}(r_3, r_4))\mathcal{P}(\Theta)} \\ =&\frac{0.3 * 0.1}{0.5 * 0.1 + 0.5 * 0.9} = 0.06. \end{aligned} \tag{2}$$

$\mathcal{P}(P_3)$ can also be obtained as the above and the final result is $\mathcal{P}(P_3) = 0.04$. As a result, the uncertainty of the possible partitions is significantly reduced from $0.464 \rightarrow 0.186$. Since $\Theta = 90\%$, this reduction in uncertainty is slightly less than what would be achieved with error-free responses. However, this demonstrates that even imperfect answers can substantially aid in diminishing uncertainty. With repeated iterations of this process, when either the entropy reduction ceases or the budget is exhausted, the more dependable distribution of possible partitions is acquired.

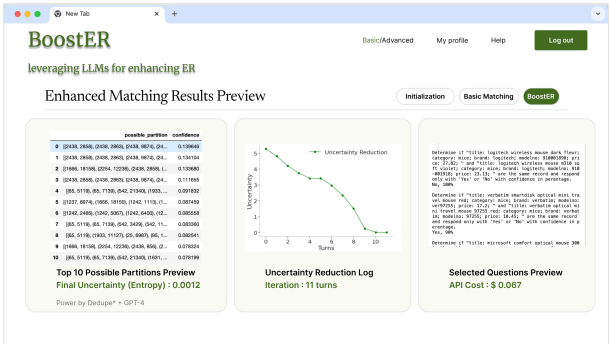


(I): Input the data and select the appropriate tools.



(II): Choose the LLM and configure your budget.

Figure 3: BoostER Demo



(III): Observe and analyze the results.

3 DEMONSTRATION

This demonstration aims to provide an interactive system for users to enhance entity resolution results by leveraging LLMs through the techniques described previously. The main three steps are depicted in Figure 3, and we will introduce them respectively.

Initialization: BoostER supports multiple data resources in .csv format, allowing for easy uploads via the "Upload" button. Currently, the system integrates built-in tools such as Dedupe [5] and Record Linkage [3]. Users can select a specific tool and initiate the process by clicking the corresponding button. Subsequently, the program automatically executes multiple iterations with various parameters.

Basic Matching: This step presents the entity resolution results produced by the fundamental tools, showcasing the top 10 possible partitions in a table format. The Initialization is designed for basic users, this implementation allows for an introduction to the system's capabilities. In the advanced version, users have the option to upload their entity resolution results. They can then select a specific LLM and establish a budget for its use.

BoostER: In the final step, the improved entity resolution results are displayed, along with the operational logs of the BoostER system. These include the uncertainty reduction curve and a log of the selected questions.

4 CONCLUSION

In this paper, we design the BoostER framework, which provides a cost-effective application of LLMs in enhancing entity resolution results. The BoostER achieves maximal effectiveness within the given budget of API request. Our target users are small companies or individual users without the need for extensive model training or significant financial investment, but can obtain high-quality results with general tools and a small amount of money. In other words, we reduce the cost of obtaining entity resolution, both in terms of ease of use and cost of use. In future work, we will explore more prompting techniques to further improve the performance.

ACKNOWLEDGMENTS

This work was partially supported by Major Program of National Language Commission (WT145-39) and Natural Science Foundation of Guangdong (2023A1515012073). And this work was also supported from the following funding sources: PolyU (UGC) - P0045695, Innovation and Technology Fund (P0043294), PolyU-MinshangCT Generative AI Laboratory (P0046453), Research Matching Grant Scheme (P0048191, P0048183), and PolyU Start-up Fund (P0046703).

REFERENCES

[1] Mikhail Bilenko and Raymond J Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 39–48.
[2] P Christen. [n. d.]. Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. 2012.
[3] J De Bruin. 2019. *Python Record Linkage Toolkit: A toolkit for record linkage and duplicate detection in Python.* https://doi.org/10.5281/zenodo.3559043
[4] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. 2006. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering* 19, 1 (2006), 1–16.
[5] Forest Gregg and Derek Eder. 2022. *Dedupe.* https://github.com/dedupeio/dedupe
[6] Di Jiang, Chen Zhang, and Yuanfeng Song. 2023. *Probabilistic topic models: Foundation and application.* Springer.
[7] 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. *arXiv preprint arXiv:2401.03426* (2024).
[8] Avanika Narayan, Ines Chami, Laurel Orr, Simran Arora, and Christopher Ré. 2022. Can foundation models wrangle your data? *arXiv preprint arXiv:2205.09911* (2022).
[9] Jiawei Tang, Yifei Zuo, Lei Cao, and Samuel Madden. 2022. Generic entity resolution models. In *NeurIPS 2022 First Table Representation Workshop.*
[10] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *arXiv preprint arXiv:1208.1927* (2012).
[11] William E Winkler. 2014. Matching and record linkage. *Wiley interdisciplinary reviews: Computational statistics* 6, 5 (2014), 313–325.
[12] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).